1989年9月1日発行(毎月1回1日発行)第8巻9号通巻89号 昭和58年11月2日第三種郵便物認可

ISSN 0910-7614

パーソナルコンピュータ・マガジン
MZシリーズ,X1/turbo,X68000&ポケコン

Oh!X

オー!エックス 定価560円

特集 活用ハードディスク&プリンタ

各社ハードディスク接続総チェック
ビデオプリンタ活用プログラム

サイバースティックで遊ぶ
不思議な環境ソフトの世界

X1/turbo
シューティングゲームDefeat X

MZ-2500
グラフィックエディタ作成講座

X68000
X-BASIC調理実習/マシン語プログラミング
C調言語講座PRO-68K/DōGA・CGA講座

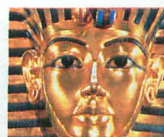
S-OS
生物進化シミュレーションBUGS

THE SOFTOUCH
ジェノサイド/琉球
mFORTH Compiler

LIVE in '89
X1/turboイタリア協奏曲/代々木ゼミナール校歌
X68000サンダークロスよりFirst Attack

猫とコンピュータ/知能機械概論
マシン語カクテル in Z80's Bar

9
SEP. 1989

夢のつづきを語ろう。

いま、ヒューマンインターフェイスなX68マインド ――
「スペックはすべてを語り尽くせない」、このパラドックスは、パーソナルな機器としてのコンピュータの難しさを端的に表現しています。"クリエイティブマインド"や"ヒューマンインターフェイス"は、ハードウェアレベルにしろ、アプリケーションレベルにしろ、スペック上で全てがわかるというものでもありません。そうした意味からも、単にハードウェアとしての32ビットには疑問をはさむ余地がありすぎるのも、また事実です。何を実現したいのか。X68000は、少なくともその可能性を提示し得た数少ないマシンのなかのひとつといえます。パーソナルデータとは何か、ビジュアルインターフェイスの意味、開発当初のコンセプトは、風化するどころか、いま現実となってますますクローズアップされてきています。かゆいところに手が届く、そんなヒューマンインターフェイスのひとつひとつをクリアーしていく、血の通ったテクノロジーに新世代マシンのイメージがふくらんできませんか。

●そのヒューマンインターフェイスを推し進めて、X68000のシステムパフォーマンスをさらに高めたものとしてHuman68k ver.2.0があります。EXPERT, PRO両シリーズに搭載されたこのOSは、従来通りのマルチウインドゥやアイコンを駆使したビジュアルシェルによるフレンドリーな操作環境に、将来性をみこした数々の処理機能を装備。インテリジェントな環境を実現しています。まず、マルチタスクに近い処理環境を提供するバックグラウンド処理の実現。バックグラウンドで動作するサンプルとして標準でTIMERコマンドを用意し、ある処理を行いながら指定時刻にAD PCMファイルを再生させたり、1画面分のメモ程度のファイルを画面に表示させることができます。次に、これからのワークステーション環境に必要なネットワーク処理にそなえ、ファイルの共有化とロックや仮想ドライブ対応などをサポートしています。さらに、キー入力や編集を効率的に行えるヒストリデバイスドライバの採用。拡張されたヒストリ機能に加え、コマンドを別の名前で定義するエイリアス機能、キー入力の履歴からユーザーが自由に複数行を登録しておき、連続実行できる簡易バッチ機能などを装備しています。その他、約2倍にスピードアップされたファイルアクセス(V1.0比)、大容量ファイルアクセスを可能にし、光磁気ディスクなど将来の大容量メディアへの対応、メニュー方式の簡単なキー操作で外部コマンドやアプリケーションを実行できるMENUコマンドの装備など、さらに高い次元へと進化した処理機能とヒューマンインターフェイス。まさにワークステーションと呼ぶにふさわしいシステムパワーを実現しました。●また、日本語処理に対してもヒューマンインターフェイスを追求した日本語フロントエンドプロセッサver.2.0を搭載。約2倍にスピードアップ(V1.0比)された変換速度をはじめ、キー割り付けの自由設定、カーソル位置での文字入力や変換など、フレンドリーなオペレーティングを実現する操作環境をサポートしています。

〈共通特長〉●プロセッサの未来を先取りした68000搭載●テキスト、グラフィック、スプライトの3画面を独立させた独自のメモリアーキテクチャ●1024×1024ドット(最大表示エリア768×512ドット)、高品位な金属の質感までも自然に表現しうる65,536色同時発色(512×512ドット時)の高解像度自然色グラフィックス●16×16ドットの緻密なキャラクタを駆使できるスプライト機能(水平32スプライト、1画面128スプライト、65,536色中16色)●ステレオFM音源、ADPCM搭載●オートロード、オートイジェクトメカ採用。インテリジェントな1Mバイト5"FDD2基搭載。●蓄積されたソフトが利用できるX68000シリーズとソフトコンパチ。

EXPERTシリーズ

●高密度実装を象徴するフォルム、マンハッタンシェイプ●新たな領域をひらく3Mバイトの大容量メモリを標準装備、メインメモリは標準で2Mバイト、最大12Mバイトまで拡張可能●プロフェッショナルなクリエイティブワークに対応する40Mバイトハードディスク搭載(CZ-612C)*●マウス・トラックボール標準装備●日本語入力にスムースに対応するASCII準拠フルキーボード

*CZ-602Cには、本体内に内蔵できる増設用の40Mバイトハードディスクドライブ(標準価格120,000円税別・取付費別)をサポート。

PROシリーズ

●意表をつくボディコンストラクション、高度な実装技術に裏付けられた洗練と信頼性の新しいスタンダードフォルム●高度なシステム化への対応を考慮した拡張I/0スロット4スロット標準装備●プロユーズの大容量ファイルに対応した40Mバイトハードディスク搭載(CZ-662C)*●2Mバイトの大容量メモリを標準装備●マウス標準装備●ワイドスケールのフルキーボード採用

*CZ-652Cには、本体内に内蔵できる増設用の40Mバイトハードディスクドライブ(標準価格120,000円税別・取付費別)をサポート。

選べる3タイプのディスプレイをサポート

15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-602D-GY(グレー)・-BK(ブラック) 標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm) CZ-612D-GY(グレー)・-BK(ブラック) 標準価格119,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ (ドットピッチ0.31mm) CZ-603D-GY(グレー)・-BK(ブラック) 標準価格 84,800円(チルトスタンド同梱・税別)

EXPERT・PRO

収獲祭 X68000見体験フェア

仙台イベントホール松栄(1階展示会場)
9/23(土)12:00〜18:00・9/24(日)10:00〜17:00

●写真左はCZ-612C-BK+CZ-612D-BK、写真右はCZ-652C-GY+CZ-603D-GY

EXEリーダーズ「カップ」プレゼント実施中
●いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズ「カップ」をプレゼントします。詳しくはEXEショップにお問い合わせください。
●また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

シャープ株式会社

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)
※本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

Oh!X

CONT

表紙絵： Moto Noriyuki

●特集

●カラー紹介

●読みもの

●シリーズ全機種共通システム

〈スタッフ〉
●編集長／前田 徹 ●副編集長／永野 仁 ●編集／植木章夫 石塚康世 高野庸一 ●協力／有田隆也 中森 章 清水和人 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／手塚喜美子 千野延明 織田洋子

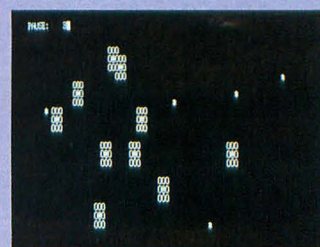
■広告目次

1989 SEP. 9

E N T S

ジェノサイド

不思議な環境ソフトの世界

Defeat X

生物進化シミュレーションBUGS

特集 スーパーワイドビデオコピー

特集 活用ハードディスク

SHARP

クリエイティブマインドあふれる周辺機器が

CZ-600C/601C/611C/602C/612C

ディスプレイ関連

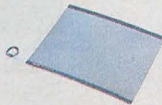
カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-GY・-BK
標準価格 99,800円（税別）
（チルトスタンド同梱）

NEW
15型カラーディスプレイテレビ
CZ-612D-GY・-BK
標準価格 119,800円（税別）
（チルトスタンド同梱）

チューナー

RGBシステムチューナー
CZ-6TU-GY・-BK
標準価格 33,100円（税別）
（リモコン付）

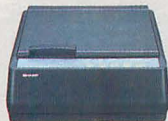
CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格 19,800円（税別）
（14/15型用）

カラーディスプレイ

21型カラーディスプレイ
CU-21CD
標準価格 139,800円（税別）

14型カラーディスプレイ
CZ-603D-GY・-BK
標準価格 84,800円（税別）
（チルトスタンド同梱）

アートツール

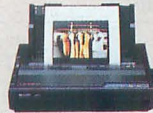
画像入力

カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円（税別）

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円（税別）

映像入力

カラーイメージユニット
CZ-6VT1
CZ-6VT1-BK
標準価格 69,800円（税別）

プリンタ

カラープリンタ

24ドット
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格 65,800円（税別）
（信号ケーブル同梱）

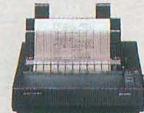
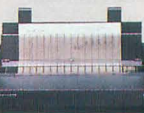
NEW
48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円（税別）
（信号ケーブル同梱）

カラービデオプリンタ

カラービデオプリンタ
★CZ-6PV1
標準価格 198,000円（税別）
（信号ケーブル同梱）

カラー イメージジェット

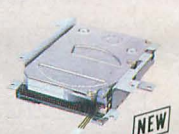
カラー イメージジェット※2 NEW
IO-735X
標準価格 248,000円（税別）
（信号ケーブル別売）

ドットプリンタ

24ピン漢字プリンタ（80桁）
CZ-8PK7
標準価格 122,000円（税別）
（信号ケーブル同梱）

24ピン漢字プリンタ（136桁）
CZ-8PK8
標準価格 152,000円（税別）
（信号ケーブル同梱）

24ピン漢字プリンタ（80桁）
CZ-8PK9
標準価格 89,800円（税別）
（信号ケーブル同梱）

ファイル

ハードディスク

ハードディスクユニット（20MB）
CZ-620H
標準価格 178,000円（税別）

NEW
増設用ハードディスクドライブ
（40MB）
CZ-64H
標準価格 120,000円（税別）
（取付費別）
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円（税別）で接続してください。
※2 別売の信号ケーブルIO-73CX標準価格5,500円（税別）で接続して下さい。
※3 CZ-652C、662Cをお持ちの方は包装箱の表示形名 CZ-6BE 1Aの右横にⒶマーク表示のあるものをお買い求めください。

X1・X1turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21CD	139,800円

映像・画像入力編集装置		
●カラー イメージスキャナ	CZ-8NS1	188,000円
●カラー イメージボードII	CZ-8BV2	39,800円
●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー（2本1組）標準装備、ミュージックツール同梱

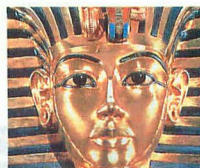
プリンタ		
●24ピン漢字プリンタ（80桁）	CZ-8PK7	122,000円
●24ピン漢字プリンタ（136桁）	CZ-8PK8	152,000円
●24ピン漢字プリンタ（80桁）	CZ-8PK9	89,800円
●24ドット熱転写カラー漢字プリンタ	CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4・-GY	99,800円
●カラービデオプリンタ	★CZ-6PV1	198,000円
●カラー イメージジェット	IO-735X	248,000円

ファイル		
●ミニフロッピーディスクユニット（2HD・2D）※3	★CZ-520F	118,000円

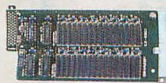
シャープ株式会社

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221（大代表）

資料請求券
X68000 ペリフェラル
oh!X
9係

X68000をサポート。

シャープペリフェラルファミリー X68000

ボード

拡張メモリ

1MB増設RAMボード（CZ-600C用）
CZ-6BE1
標準価格 35,000円（税別）

1MB増設RAMボード※3（CZ-601C/611C/652C/662C用）
CZ-6BE1A
標準価格 38,000円（税別）

2MB増設RAMボード※4
CZ-6BE2
標準価格 79,800円（税別）

4MB増設RAMボード※4
CZ-6BE4
標準価格 138,000円（税別）

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円（税別）

GP-IBボード
CZ-6BG1
標準価格 59,800円（税別）

増設用RS-232Cボード（2チャンネル）
CZ-6BF1
標準価格 49,800円（税別）

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円（税別）

FAX

FAXボード
CZ-6BC1
標準価格 79,800円（税別）

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円（税別）

ネットワーク

モデム

モデムユニット※5
CZ-8TM2
標準価格 49,800円（税別）
（RS-232Cケーブル同梱）

RS-232Cケーブル

RS-232Cケーブル（平行接続型）
CZ-8LM1
標準価格 7,200円（税別）

RS-232Cケーブル（クロス接続型）
CZ-8LM2
標準価格 7,200円（税別）

入力

インテリジェントコントローラ NEW
CZ-8NJ2
標準価格 23,800円（税別）

マウス・トラックボール NEW
CZ-8NM3
標準価格9,800円（税別）

トラックボール
CZ-8NT1
標準価格 13,800円（税別）

マウス
CZ-8NM2A
標準価格 6,800円（税別）

ジョイカード
CZ-8NJ1
標準価格 1,700円（税別）

その他

拡張スロット

拡張I/Oボックス（4スロット）（CZ-600C/601C/611C/602C/612C用）
CZ-6EB1
CZ-6EB1-BK
標準価格 88,000円（税別）

スピーカー

アンプ内蔵スピーカーシステム（2本1組）
AN-S100
標準価格 36,600円（税別）

システムラック

システムラック
CZ-6SD1
標準価格 44,800円（税別）

※4 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円（税別・CZ-600C用）、CZ-6BE1A 標準価格38,000円（税別・CZ-601C、CZ-611C、652C、662C用）を増設してください。
※5 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

品名	型番	価格
●ミニフロッピーディスクユニット（2D）	★CZ-502F	99,800円
●ミニフロッピーディスクユニット（2D・1ドライブ）	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ（2D）※4	CZ-53F-BK	19,800円

拡張ボード・その他		
●モデムユニット（300/1200ボー）	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円

品名	型番	価格
●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円
●RS-232C用ケーブル（平行接続型）	CZ-8LM1	7,200円
●RS-232C用ケーブル（クロス接続型）	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

品名	型番	価格
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド※9	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター※10	BF-68PRO	19,800円
●スキャナ用パラレルボード※11	CZ-8BN1	27,800円

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 CZ-600D、880D、830D用 ※10 14/15型用 ※11 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。
をご参照ください。

★印の商品は在庫僅少です。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161（大代表）

SHARP

"アート"と呼べる高水準のソフトウェアが

必要なとき、いつでも使える、サッと呼び出せる。
メモリ常駐型の便利ツール。
いまトレンディなステーショナリーソフトウェア。

文房具感覚で使えるサポートツール

Stationery PRO-68K

CZ-240BS　　8月発売予定

パソコンを本当に道具として、現実的に使いこなしたい、そんなユーザーのためのソフトウェアが登場しました。「Stationery PRO-68K」、それはパソコンの使い勝手を飛躍的に向上させる便利ツールです。

●他のソフトウェアを実行中でも呼び出して使えるメモリ常駐型のソフトウェア。

使い方は簡単、他のアプリケーションを起動する前に、この「Stationery PRO-68K」を一度起動するだけ。これで他のアプリケーション実行中にも、「メモ」や、「スケジュール」、「住所録」など、「Stationery PRO-68K」の持つ機能がワンタッチで使えます。いちいちアプリケーションを終了させることもありません。

●シャープ電子手帳のデータをX68000で入力、編集。

パソコン上で入力したデータを電子手帳の「電話帳」、「スケジュール」、「メモ」へデータを送信したり、逆に電子手帳側からデータを受信して編集することができます。(別売の通信ケーブルが必要です。)

メモ ●先頭/最終ページへのジャンプ●文字列の置換、検索●ファイルの読み込み/書き出し、などの機能をもつ小型エディタ。

カレンダー ●任意の半年分のカレンダー表示可能。

スケジュール ●日付/時間/メモという3つの項目を持つ表形式●入力したデータは日付順に自動整列●先頭/最終ページへのジャンプや文字列検索機能を装備。

住所録 ●氏名/索引/電話/住所の4つの項目をもつ表形式●先頭/最終ページへのジャンプ、項目ごとの整列、文字列検索、重複データ消去、ファイルの読み込み/書き出し、などの機能を装備。

シャープ株式会社　●お問い合わせは…シャープ㈱電子機器事業本部テレビ事業部 第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券 CZソフト oh!X 9係

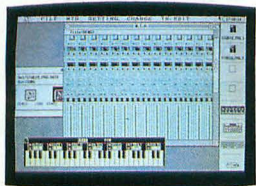
X68000をサポート。

シャープオリジナルソフトウェア X68000

サウンドツール

Musicstudio PRO-68K
■CZ-237MS 標準価格25,800円(税別)

24の録音トラックをもったプロフェッショナルユースのMIDIマルチレコーディングソフトです。MIDI楽器を使って演奏したデータをスタジオ感覚で編集し、記録、再生できます。MUSIC PRO-68KのMMLデータコンバートも可能です。

※MIDIボード(CZ-6BM1)が必要です。

MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

MIDI対応自動伴奏機能をサポート、簡単な楽譜入力で演奏が楽しめます。

※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工作成用からなるライブラリです。

Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ&演奏用ツール。

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

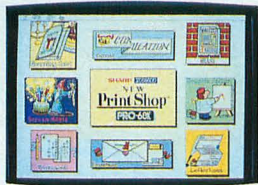
スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。

グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

暑中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

ビジネスツール

TOP給与計算エキスパート
■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計
■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

コマンド入力の手間を軽減するヒストリー機能、罫線ドライバー付レポートライター機能、10進31桁の高精度演算。さらにイメージ表示機能を装備したコマンド型リレーショナルデータベースです。

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を緊密に一体化させた統合ビジネスツールです。マウス対応のやさしいオペレーション、高度なエディタ機能、豊富な関数群など、初心者からプロまで幅広く使えます。

通信ツール

Communication PRO-68K
■CZ-223CS 標準価格19,800円(税別)

300～19,200BPSまでの通信速度に対応し、各種データベースの漢字端末やパソコン通信に利用できます。逆スクロール機能、自動実行機能、コンカレント機能も装備。さらに豊富な編集機能をもった高機能通信ソフトです。

開発ツール

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

X68000のもつグラフィック環境はもちろん、AD PCM音声、FM音源とグラフィックの同時再生といったマルチメディア機能をサポート。OS-9のもつマルチタスク機能、リアルタイム機能を活かした使い易く機能的なOS環境を提供します。また、これまでのデータ資産も活かせます。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

システムパフォーマンスを高める処理機能を付加したHuman 68kの最新バージョンです。マルチタスクに近い処理環境を提供するバックグラウンド処理、ネットワーク処理、ファイルアクセスのスピードアップなど、さらに高い次元へと進化した機能とユーザーインターフェイス。大容量メディアにも対応。

C compiler PRO-68K
■CZ-211LS 標準価格39,800円(税別)

Cコンパイラ、BASIC-Cコンバータ、アセンブラ、リンカ、デバッガ、アーカイバ、コンバータからなるツール。OS上のプログラム開発を効率良くサポートします。XCはC言語の基本的な仕様に準拠し、ANSI仕様も採用、ハードウェアをサポートした豊富なライブラリ(約700種)も用意されています。

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

アセンブラ、リンカ、デバッガ、アーカイバ、X-BASIC V2.00からなる手軽な開発ツールです。

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS 標準価格188,000円(税別)

AI開発用言語とエキスパート構築ツールがセットになったAIプログラム開発ツールです。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

ドラゴン

DRAGON

X68000で衝激にデビュー!!

今、流りのパズルゲーム「DRAGON」がいよいよX68000に登場。盤上に並べられた麻雀牌をルールに従って全て裏返せば一面クリア。クリアした後には、ユニークなアニメーションを展開。多彩なファイル機能やランダム処理により、奥深いゲームが楽しめる。

※MZ-2500シリーズも好評発売中。

X68000版 DRAGONの仕様。

- ▶新たに描き直したX68000専用のグラフィックス。
- ▶ADPCMによる鮮明な音声出力機能。
- ▶多彩に用意されたゲーム完成後のアニメーション。
- ▶セレクトモード付きのステレオサウンド。

X68000

HIGH RESO. TIMER POWER

定価6,800円（税別）

コンピュータシステム・ソフトウェア企画制作

株式会社

〒110 東京都台東区台東2-4-3松本ビル4F ☎03-837-2595

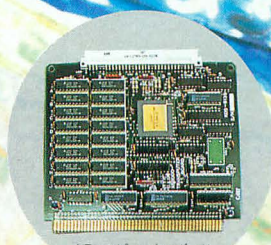
★フルカラーフレームバッファ、ペイントソフトセット新発売//〈X-68000用、PC-9801用〉

株式会社 キャスト
〒158 東京都世田谷区等々力2-1-13
TEL.03-705-0656 FAX.03-705-5224

CasT

製品	価格
C-TRACE TOWNS	¥68,000
C-TRACE 68 (X68000対応)	¥68,000
C-TRACE 98DRY (PC-9801対応)	¥68,000
C-TRACE 98+ (PC-9801対応)	¥198,000
C-TRACE NEWS (SONY)	¥380,000
★C-TRACE 98TP	¥610,000
★C-TRACE 68TP	¥610,000

表示価格に消費税は含みません
★の製品は店頭販売いたしません。直接当社までお申し込みください。

▲このCGは、千葉県山川秀幸さんの作品です。

トランスピューターボード

この夏、うれしいはやさです。

新登場、超高速レイトレーシング
C-TRACE TP新発売!!
トランスピューターの使用により、
動作速度はグーンとアップ。
X68000の
なんと、170倍で新登場です。
この速さと美しさをぜひお試しください。

★「C-TRACE教室」開講中。

市ヶ谷シャープ(株)内OAショールーム
にて開催中。詳しくは、キャストまで。
講師:
長谷川一光、宮嶋美奈子(CG作家)

★C-TRACEによるTVCM作品集(VHSのみ)をお分けいたします。
代金1,000円を現金書留にてお送りください。送料着払いにて発送いたします。

プロのための3次元コンピューター グラフィックス

C-TRACE

商標登録申請中

Multi Visual Weapon

PRISM 68K

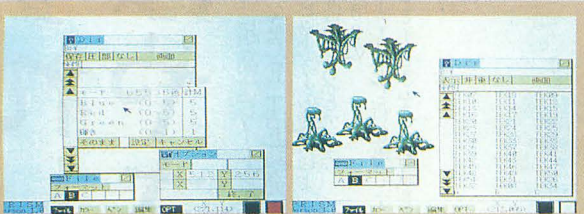
9月1日(金)全国で発売開始
プリズム68K ¥38,000(税抜き)

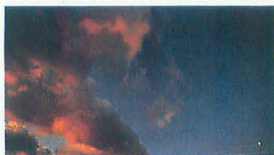
ツールはウェポンに進化する――。

―必要のためのマルチ・モード環境―

■欲しい解像度、必要な色数のために2401通りのヴィジュアル環境を提供。(グラフィック・モードは、512×512、512×256、256×512、256×256の4モード。カラー・モードは、65536色、256色、64色、16色、8色、4色、2色の7モード。)

―プロ仕様のための、ウェポン環境―

■データのセーブ形式を、マニュアルにて詳細に解説。
■各グラフィック・ツールのデータをロードするオプションを装備。
■簡易スプライト・エディターを搭載。
■256色以下のカラー・モードでは、色チェンジに対応。
■任意の場所に任意の多重ポップ・アップ・ウィンドウ。

X68000シリーズ好評発売中!! (価格に消費税は、含まれません)

¥10,000
ニュータイプ・ドラマチック・RPG
ARCUS pro68K

¥12,800
ハイパー・シューティング・アクション
MID-GARTS gold68K

¥9,800
ハードボイルド・ADV
GAUDI-バルセロナの風-68K

ゲーマーズ・ホット・アクセス TEL03(5273)4795

※通信販売ご希望の方は、商品名、機種名、住所、氏名、電話番号を明記の上、現金書留にてお申込ください。(送料無料)
※当社は当社が著作権を有する本ソフトウェアのレンタル行為、及び複製行為について、これを一切許可しておりません。もし違反した場合は懲役または罰金が課せられます。

■適応機種：X68000
定価 **¥9,800**
(5"2HD 4枚組)

好評発売中！

トリトーン ファイナル TRITORN FINAL

- 爆発的人気を誇る「トリトーン」の決定版！
  X68000専用バージョン、初のアクション、R.P.G.。
- 「トリトーン2」オリジナルシナリオを超える数々の謎とイベント。そして、増大なアイテム数と魔法数。
- 魔法効果をよりビジュアルに表現させた抜群のアクション性能。
- オールタイム・フルカラー(512×512、256色)二重マップクロールのX68000専用グラフィック。
- 6000以上にも及ぶ会話の数々!!
- ミュージックファン待望のオリジナルB.G.M.。

僕は夏の夜に雪国をさまよい歩く…。

8月下旬発売予定

■適応機種：X68000
定価 **¥8,800**
(5"2HD 3枚組)

- 「クルージュ」オリジナルシナリオに多数のイベントを追加し(X68000版専用バージョン)ムービング・アドベンチャー・ゲーム新登場！
- 目を見張る素晴らしいグラフィックに広大なマップ(512×512、256色)
- 更に奥深い謎の連続、そして次々と明らかになる新事実！
- ストーリー重視の新しいゲームジャンルのムービング・アドベンチャー。
- 多数のオリジナルB.G.Mに加えて、ADPCM(音声合成)による豊富な効果音。
- マウス対応。

グラフィック・
スプライト・
キャラクターエディター

G-ツール

■適応機種：X68000

■アセンブラ・C言語でオリジナルゲーム(X68000の機能をフルに活かした)をこれから制作しようと思う人に使っていただきたい本格派ツールの登場です。

- 本格派スプライトエディター
- フルカラー・グラフィックエディター
- キャラクターエディター
（マウス対応・全グラフィックモード対応）

予価 **¥28,000** 発売日未定

※表示価格には消費税は含まれておりません。
※通信販売ご希望の方は、商品名、機種名、住所、電話番号を明記の上、現金書留にてお申し込み下さい。(送料サービス)

- スタッフ募集！
  プログラマー・デザイナーを募集しております。お問い合せください。

制作：リー・ウェイ・コーポレーション
販売：株式会社ザイン・ソフト
〒676 兵庫県高砂市米田町米田1162-1 TEL.(0794)31-7453

■冬のボーナス一括払いOK.!! 手数料ナシ!! 12月末払いOK.!! おトクですネェ、ぜひ!!

パソコンプラザ

OCT オクト

案内図

JR蒲田から徒歩5分
京浜蒲田から徒歩1分

店頭セール実施中

オクトで始まるパソコンワールド

☎ 03-730-6271

●営業時間 AM 11:00〜9:00/日曜・祭日PM7:00 電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販 ●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

12回	4.5%	24回	10%	36回	14%	48回	18%

OCT-1 システム インフォメーション

▶全商品保証付(メーカー保証)
▶超低金利ハッピークレジット(1回〜60回)頭金ナシOK!
▶ボーナス一括払いOK! ボーナス2回払いOK.!!
▶配達日の指定OK!(万全なサポート体制)
▶商品の組合せ自由! オクトフリーダムシステム
▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

OCT-1 オクト-1 蒲田

残暑お見舞申し上げます!!
ラストサマーチャンスの贈りものです、ハイ!!
X68000フェア開催中!!

OPEN

《新製品発売記念プレゼント実施中》★セットでお買い上げの方には、アフターバーナー(¥9,200)をプレゼントいたします。

お好みのセットをお選び下さい。 送料無料

●3Mバイトの大容量メモリ
●40Mバイトハードディスク搭載

EXPERT・EXPERT-HD

●CZ-602C(BK) 定価¥356,000
●CZ-612C(BK) 定価¥466,000

現金特価!! 推選 お電話下さい。

●拡張I/Oポート4スロット装備
●2Mバイトの大容量メモリ

PRO・PRO-HD

●CZ-652C(GY/BK) 定価¥298,000
●CZ-662C(GY/BK) 定価¥408,000

CZ-8NJ2
●インテリジェントコントローラ
定価¥23,800
超特価!お電話下さい。

15型カラーディスプレイTV

CZ-612D-GY/BK NEW
定価¥119,800

15型カラーディスプレイTV

CZ-602D-GY/BK NEW
定価¥ 99,800

14型カラーディスプレー

CZ-603D-GY/BK
定価¥84,800

21型カラーディスプレイ

CU-21CD
定価¥139,800

Ⓐ CZ-602C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥475,000▶ウフフ。お買徳ですヨ!
Ⓑ CZ-612C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥585,800▶超低金利クレジットをご利用下さい。
Ⓒ CZ-652C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥417,800▶電話一本。ハイ即納。
Ⓓ CZ-662C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥527,800▶超特価!電話下さい。

Ⓔ CZ-602C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥455,800▶超特価!電話下さい。
Ⓕ CZ-612C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥568,800▶ウフフ。お買徳ですヨ!
Ⓖ CZ-652C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥397,800▶超低金利クレジットをご利用下さい。
Ⓗ CZ-662C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥507,800▶電話一本。ハイ即納。

Ⓘ CZ-602C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥440,800▶電話一本。ハイ即納。
Ⓙ CZ-612C＋CZ-603D＋MD-2HD20枚＋ゲーム ………定価¥550,800▶超特価!電話下さい。
Ⓚ CZ-652C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥382,800▶ウフフ。お買徳ですヨ!
Ⓛ CZ-662C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥492,800▶超低金利クレジットをご利用下さい。

Ⓜ CZ-602C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥495,800▶超低金利クレジットをご利用下さい。
Ⓝ CZ-612C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥605,800▶電話一本。ハイ即納。
Ⓞ CZ-652C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥437,800▶超特価!電話下さい。
Ⓟ CZ-662C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥547,800▶ウフフ。お買徳ですヨ!

※クレジットの回数は1回〜60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料¥2,000 ●店頭デモ実施中…専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

オクトで始まるパソコンワールド。店頭にて、ゲームソフト25%OFF!!(税別)超低金利ハッピークレジットをご利用ください。

■冬のボーナス一括払いOK.!! 手数料ナシ!! 12月末払いOK.!! おトクですネェ、ぜひ.!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

ラストチャンス! X68000ACE-HD超特価セール!!

※セットでお買上げの方にはアフターバーナー(ゲーム)をプレゼント.!!

限定 送料¥2,000

X68000 ACE-HD

推奨セット

㊙超特価 絶対! お徳デス!!

Ⓐ CZ-611C+CZ-603D+MD-2HD+ゲーム ……▶超特価!TEL下さい。

Ⓑ CZ-611C+CZ-602D+MD-2HD+ゲーム ……▶超特価!TEL下さい。

Ⓒ CZ-611C+CZ-611D+MD-2HD+ゲーム ……▶超特価!TEL下さい。

Ⓓ CZ-611C+Cu-21CD+MD-2HD+ゲーム ……▶超特価!TEL下さい。

なんとウレシイ! X-1G超特価(送料無料)

限定

X-1G(本体)

- CZ-882C
- MD-2HD10枚
- ジョイカード(連射)
- ゲームソフト1本

買わなきゃソンをする!! 早い者勝ち!!

㊢大特価¥29,000

※超低金利クレジットご利用下さい。1回〜60回払い、頭金ナシ!ボーナス1回払い、ボーナス2回払いOK.!

型名	商品	特価	特価	型名	商品	定価	特価
CZ-6BE1	1MB増設RAMボード	¥ 38,000	大特価	CZ-6EB2	拡張I/Oボックス	¥ 88,000	大特価
CZ-6BE2	2MB増設RAMボード	¥ 79,000	大特価	CZ-8TMZ	モデムユニット	¥ 49,800	大特価
CZ-6BG1	GP-IBボード	¥ 59,800	大特価	CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	大特価
CZ-6BP1	プロセッサ・ボード	¥ 79,800	大特価	CZ-8NT1	トラックボール	¥ 13,800	大特価
CZ-6BC1	FAXボード	¥ 79,800	大特価	CZ-6BU1	ユニバーサルI/Oボード	¥ 39,800	大特価
CZ-6BM1	MIDボード	¥ 26,800	大特価	AN-160SP	アンプ内蔵スピーカ	¥ 59,800	大特価
AN-8TV	パソコンチューナー	¥ 35,800	大特価	CZ-6PV1	カラービデオプリンタ	¥198,000	大特価
CZ-8NS1	カラーイメージスキャナ	¥188,000	大特価	CZ-6VT1-BK	カラーイメージユニット	¥ 69,800	大特価

熱転写カラー漢字プリンター 用紙プレゼント 送料無料

CZ-8PC4 ¥99,800

- 48ドットサーマルヘッド
- B5〜B4まで
- ハガキ可能
- カラー対応

大特価 オクト推選 TEL下さい!

①CZ-8PK7(24ピン80桁) 定価¥122,000····大特価・TEL下さい。

②CZ-8PK8 (24ピン136桁) 定価¥152,000····大特価・TEL下さい。

③CZ-8PK9 定価¥89,800····大特価・TEL下さい。

④CZ-8PC3(24ドット漢字カラー) 定価¥65,800····大特価・TEL下さい。

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる棚板5段のマルチに活用できるディスク。ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥16,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥12,000

X68000ソフト大セール実施中 ※ゲームソフトオール25%off

<グラフィック>● Z's STAFF PRO68K Ver.2.0
(シャフト)定価¥58,000
オクト特価¥41,000

<データベース>● KAMIKAZE
(サムシンググッド)¥定価68,000
オクト特価¥47,000

<グラフィック>● C-TRACE68
(キャスト)定価¥68,000
オクト特価¥51,000

<C言語>● C & Professional Pack
(マイクロウェアジャパン)定価¥58,000
オクト特価¥44,000

<グラフィック>● サイクロン エキスプレス
定価¥78,000
オクト特価¥58,000

型名	商品	定価	特価
BUSINESS PRO68K	統合型表計算	¥68,000	大特価
CARD PRO68K	カード型データベース	¥29,800	大特価
DATA PRO68K	コマンド型データベース	¥58,000	大特価
COMMUNICATION PRO68K	通信ソフト	¥19,800	大特価
OS-9 X68000	マルチタイム リアルタイム オペレーティング システム	¥29,800	大特価
MUSIC PRO68K	楽譜ワープロ	¥18,800	大特価
SOUND PRO68K	サウンドエディタ	¥15,800	大特価
NEW PRINT SHOP PRO68K	ポップアートツール	¥19,800	大特価
C-COMPILER PRO68K	Cコンパイラ	¥39,800	大特価
EW	ワープロ	¥38,000	¥29,800
G-68	グラフィックツール	¥14,800	¥12,000
E-68K	スプライトエディタ	¥19,800	¥16,000

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

●尚、送料として1ケ¥500、2ケ¥700、3ケ以上で¥1,000となります。(税別)

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。● 入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

低金利クレジットをご利用下さい
オクトラクラククレジット

12回	4.5%	24回	10%	36回	14%	48回	18%

振込先
富士銀行 久ヶ原支店 ㊂No.1824
三菱銀行 蒲田支店 ㊂No.0278691
株式会社 億人(オクト)

※掲載の価格は7/20現在ですので、まずは、お電話にてご確認ください。8/22・23は連休とさせていただきます。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。

※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

オクトで始まるパソコンワールド。

店頭にて、ゲームソフト25%OFF!!(税別)超低金利ハッピークレジットをご利用ください。

超低金利クレジットOK!!

またまた秋葉原でおなじみの

注目!!
冬のボーナス一括払いOK!!
手数料(金利)無料
(12月末払い、ご利用下さい)

8/15〜9/20

●お近くの方はお
●本体単品で特
●ビジネスソフト定

CYBER STICK
●CZ-8NJ2
(定価¥23,800)
超特価!!
▶価格はTEL下さい

X-1ターボZIII 特別ご提供品!! 台数限定
●CZ-888C+CZ-880D+M-2HD(10枚)
定価¥269,600▶超特価TEL下さい!!
(ジョイカード ゲーム3種 プレゼント!!)

12回	16,900	24回	8,800	36回	6,100	48回	4,700	60回	3,900

ジョイスティック 送料¥500
●X-1PRO
定価¥9,500▶特価¥7,800
●ASCII STICK
定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用くださいИ!!

X68000EXPERT & EXPERT-HD (送料¥2,000)

シャープ
CZ-8PC2
定価¥69,800
●熱転写プリンター
●48ドット

Ⓐセット:CZ-602C+CZ-603D+M-2HD(10枚)+ゲーム
…………定価¥440,800▶P&A超特価(現金価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-612C+CZ-603D+M-2HD(10枚)+ゲーム
…………定価¥550,800▶P&A超特価(現金価格はお電話下さい)

12回	?	24回	?	46回	?	48回	?	60回	?

※モニターをCZ-612D(¥119,800)、CZ-602D(¥99,800)、CU-21CD(¥139,800)に変更の場合も超特価で販売しております。TEL下さい。

※X68000セットでお買い上げの方に
アフターバーナー(定価¥9,200)をプレゼント!!

X68000PRO & PRO-HD (送料¥2,000)

※プリンターなしの組合せ
※他のプリンターの組合せ
もあります。お電話下さい。

Ⓒセット:CZ-652C+CZ-603D+M-2HD(10枚)+ゲーム
…………定価¥382,800▶P&A超特価(現金価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓセット:CZ-662C+CZ-603D+M-2HD(10枚)+ゲーム
…………定価¥492,800▶P&A超特価(現金価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

※モニターをCZ-612D(¥119,800)、CZ-602D(¥99,800)、CU-21CD(¥139,800)に変更の場合も超特価で販売しております。TEL下さい。

※X68000セットでお買い上げの方に
アフターバーナー(定価¥9,200)をプレゼント!!

X68000ACE-HD〜P&Aスペシャルセット=限定誌上販売!!

X-68000ACE-HDセット(台数限定)

●CZ-611C(本体)
●CZ-603D(モニター)
●CZ-8NJ2(CYBER STICK)
+
●ディスケット10枚
●ゲーム
●送料、消費税込み
⇒ 定価¥508,400
P&A超特価
価格はお電話下さい

12回	28,700	24回	15,000	36回	10,300	48回	8,000	60回	6,700

Ⓐのモニターを CZ-602D(定価¥99,800)に変更の場合

12回	30,100	24回	15,700	36回	10,800	48回	8,400	60回	7,000

●CZ-612D(定価¥119,800)に変更の場合

12回	31,300	24回	16,400	36回	11,300	48回	8,700	60回	7,300

●CZ-611D(定価¥145,000)に変更の場合

12回	30,700	24回	16,100	36回	11,000	48回	8,600	60回	7,100

(ボーナス併用も有ります。TEL下さい。)

●本広告の掲載の商品の価格については、消費税は含まれておりません。4月1日以降より消費税が付加されますので、ご了承下さい。
●お知らせ 5月21日より営業時間の変更=平日AM10:00〜PM8:00、日祭AM10:00〜PM7:00

回～60回払いまでOK!!

★頭金なし!★即日発送

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

商品名	定価	特価
Z's STAFF PRO68K Ver2.0(ツァイト)	定価¥ 58,000	特価¥ 40,600
C-TRACE68(キャスト)	定価¥ 68,000	特価¥ 50,300
彩CRONE(アンス・コンサルタンツ)	定価¥ 58,000	特価¥ 44,600
アニメキット(アンス・コンサルタンツ)	定価¥ 5,000	特価¥ 4,000
テラッツォ(ハミングバード)	定価¥ 19,800	特価¥ 15,800
G-68K(OH! BISINESS)	定価¥ 14,800	特価¥ 11,400
KAMIKAZE(サムシング・グッド)	定価¥ 68,800	特価¥ 46,800
EW&EI(イースト)	定価¥ 38,800	特価¥ 28,800
C&Professional Pack(マイクロウェアジャパン)	定価¥ 58,800	特価¥ 46,000
Final Ver3.2(エーエスピー)	定価¥ 38,000	特価¥ 30,000
DATA PRO68K CZ220BS	定価¥ 58,000	P&A特価
CARD PRO68K CZ226BS	定価¥ 29,800	TEL下さい。!
C compiler PRO68K CZ211LS	定価¥ 39,800	特価¥ 32,000
OS-9/X68000 CZ219SS	定価¥ 29,800	P&A特価 TEL下さい。
AI-68K CZ234LS	定価¥188,000	特価¥143,000
THE福袋V2.0 CZ224LS	定価¥ 9,980	特価¥ 18,000
SOUND PRO68K	定価¥ 15,800	特価¥ 12,500
MUSIC PRO68K CZ213MS	定価¥ 15,800	P&A特価 TEL下さい。
Sampling PRO68K CZ215MS	定価¥ 17,800	特価¥ 14,000
MUSIC-studio PRO68K 237MS	定価¥ 15,800	P&A特価 TEL下さい。
MUSIC-PRO68K〔MIDI〕 247MS	定価¥ 18,800	特価¥ 22,000
New-print Shop 221HS	定価¥ 19,800	P&A特価
Communication 223CS	定価¥ 19,800	TEL下さい。!

ゲームソフト(1ヶ～20ヶまで送料¥500)

X68000用

	商品名	定価	特価
Ⓐ	源平討魔伝(電波新聞社)	定価¥ 7,800	特価 ¥ 6,200
Ⓑ	ドラゴンスピリット(電波新聞社)	定価¥ 8,800	特価 ¥ 7,000
Ⓒ	スペースハリアー(電波新聞社)	定価¥ 6,800	特価 ¥ 5,400
Ⓓ	熱血高校ドッジボール部(SHARP)	定価¥ 7,800	P&A超特価
Ⓔ	沙羅曼蛇(SHARP)	定価¥ 8,800	P&A超特価
Ⓕ	フルスロットル(SHARP)	定価¥ 8,800	P&A超特価
Ⓖ	琥珀色の遺言(リバーヒルソフト)	定価¥ 9,800	特価 ¥ 7,800
Ⓗ	ザ・スーパーラスベガス(日本デグスタ)	定価¥ 12,800	特価 ¥10,200
Ⓘ	マイト・アンド・マジック(スタークラフト)	定価¥ 9,800	特価 ¥ 7,800
Ⓙ	ザ・リターン・オブ・イシター(SPS)	定価¥ 7,800	特価 ¥ 6,200
Ⓚ	信長の野望(全国版)(KOEI)	定価¥ 9,800	特価 ¥ 7,800
Ⓛ	麻雀悟空(シャノアール)	定価¥ 7,800	特価 ¥ 6,200
Ⓜ	マーダークラブDX(リバーヒルソフト)	定価¥ 7,800	特価 ¥ 6,200
Ⓝ	ザキングオブシカゴ(ボーステック)	定価¥ 12,800	特価 ¥10,200
Ⓞ	今夜も朝までパワフルまあじゃん2(dB-SOFT)	定価¥ 7,800	特価 ¥ 6,200
Ⓟ	三国志(光栄)	定価¥ 14,800	特価 ¥12,000

モデムコーナー (送料¥1,000)

	商品名	定価	特価
Ⓐ	MD-2400B(オムロン)	定価¥49,800	特価¥36,000
Ⓑ	MD-2400F(オムロン)	定価¥59,800	特価¥42,000
Ⓒ	PV-A2400MNP4(アイワ)	定価¥46,800	特価¥35,000
Ⓓ	PV-A24MNP5(アイワ)	定価¥54,800	特価¥41,000

カラービデオプリンター (送料¥1,000)

Ⓐセット:CZ-6PVI……定価¥198,000→**TEL下さい**

12回	13,400	24回	7,000	36回	4,800	48回	3,700

P & A 特選パソコンラック (送料無料)移動自由(キャスター付)

Ⓐ3段 875(H)×580(D)×610(W) ¥9,000
Ⓑ4段 1320(H)×600(D)×630(W) ¥12,000
Ⓒ5段 1280(H)×600(D)×620(W) ¥16,500

カラーイメージスキャナ (送料¥1,000)⊕ジェット

Ⓐ CZ-8NSI…定価¥188,000→P&A超特価!!
Ⓑ JX-100…定価¥ 89,800→P&A超特価!!
Ⓒ JX-200…定価¥198,000→
Ⓓ 10-730…定価¥230,000→P&A超特価!!
Ⓔ 10-735…定価¥248,000→P&A超特価!!

周辺機器コーナー(送料¥1,000)●その他の周辺機器はお電話下さい。

	商品名	定価	特価
Ⓐ	CZ-8BSI(FM音源ボード)	定価¥23,800	P&A超特価
Ⓑ	CZ-8RLI(データレコーダ)	定価¥24,800	特価¥20,000
Ⓒ	CZ-6BE1A(1M RAM)	定価¥38,000	P&A超特価
Ⓓ	CZ-6BE4(4M RAM)	定価¥138,000	P&A超特価
Ⓔ	CZ-6BP1(数値演算)	定価¥79,800	特価¥61,000
Ⓕ	CZ-6VTI(カラーイメージユニット)	定価¥69,800	P&A超特価
Ⓖ	CZ-6EBI(拡張I/Oボックス)	定価¥88,000	特価¥69,000
Ⓗ	AN-160SP(アンプ内蔵スピーカーシステム)	定価¥59,800	特価¥47,000

中古パソコン 送料¥2,000

商品名	価格	商品名	価格	商品名	価格
● X-68000セット	¥220,000	● CZ-856C	¥55,000	● CU-14AG2	¥40,000
● X-68000ACEセット	¥250,000	● CZ-870C	¥65,000	● CU-14H2	¥40,000
● X:1ターボZセット	¥110,000	● CZ-881C	¥75,000	● CZ-8PC2	¥35,000
● X-1G/30セット	¥ 49,000	● CZ-820D	¥20,000	● CZ-8PK6	¥42,000
● CZ-822C	¥ 25,000	● CU-14GB	¥15,000		
● CZ-830C	¥ 35,000	● CU-14BD	¥35,000		

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884
FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～60回払いまで出来ます。但し、1回のお支払い額は3,000円以上。

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

超低金利クレジット率

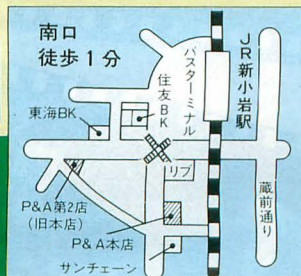
回数	1	3	6	10	12	15	18	24	36	48	60
利率(%)	1.5	2.0	3.0	4.5	4.5	7.5	9.0	9.5	13	17	22

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

●マイコン
●ビデオ
●ビデオテープ

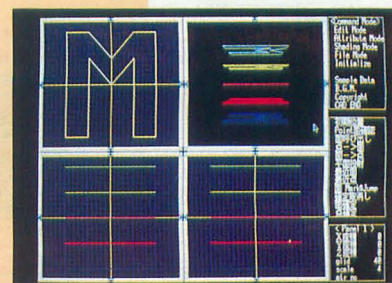
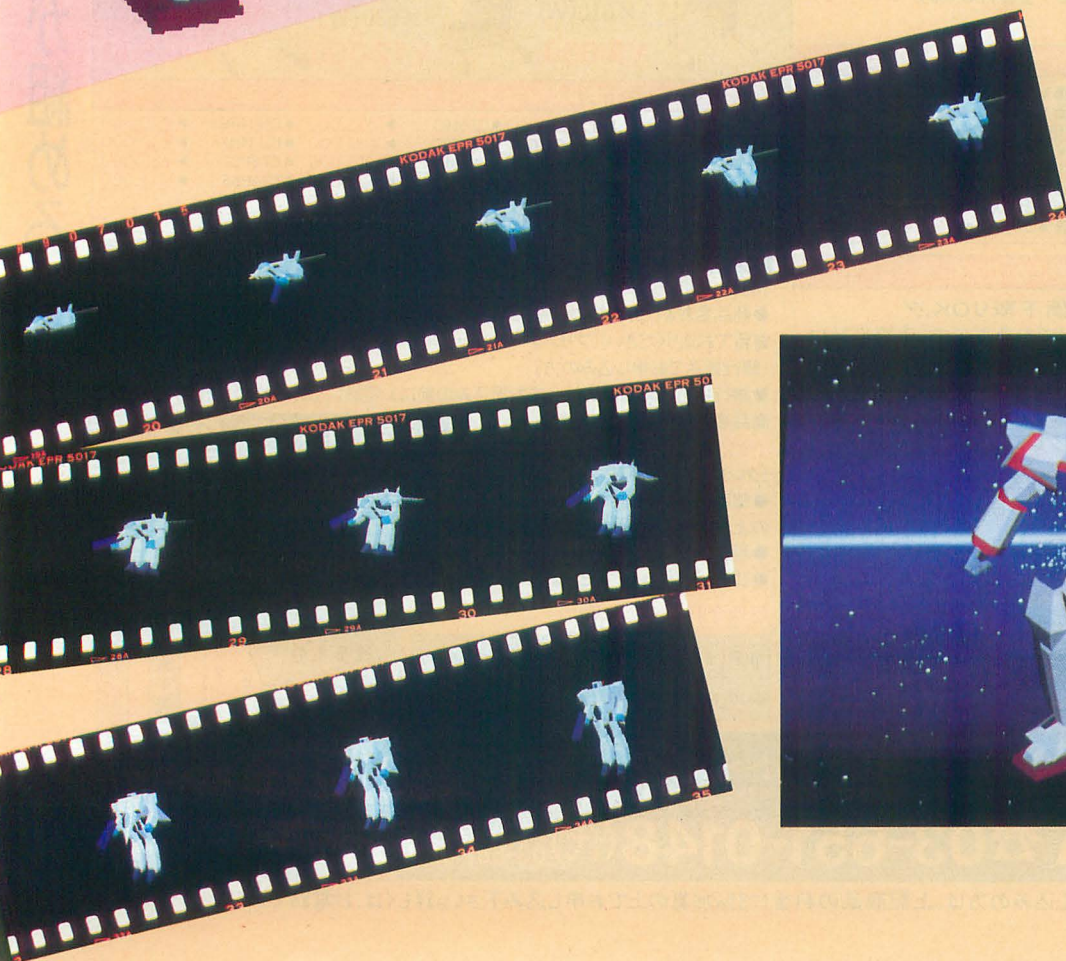
P&A

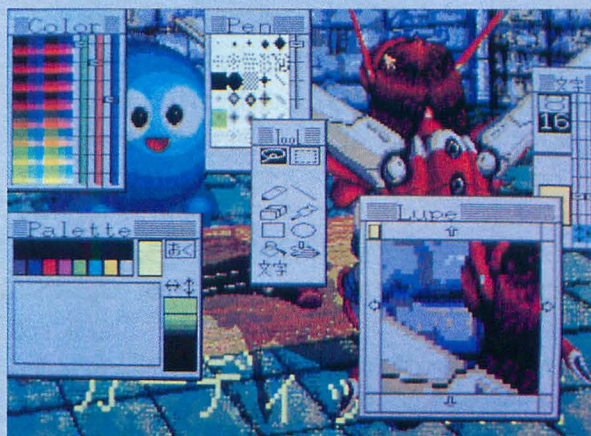
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX.03-651-0141

営業時間
平日AM10:00～PM8:00
日祭AM10:00～PM8:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

Oh!X Graphic Gallery

DōGA・CGアニメーション講座

おー，いよいよDōGAのCGアニメーションが本格的に始動し始めたぞ！ネーミングは安易だが，ついに基本システム“CAD”も今回ベールを脱ぐぞ！ここで紹介するのはDōGAのメンバー高津正道さんのロボットアニメーション2作品。上のロボットは指の関節までちゃんと作ってある。右下はCADの画面。

MZ-2500グラフィックエディタ作成講座

今月は，上下・左右反転を可能にするトランスフォーム機能と，文字描画のパラメータを設定するウィンドウの登場だ！　これで，画面上に文字を書き込むことができるようになった。オマケに，この「トランスフォーム誕生秘話」も今回は聞けたりするのだ。

特集　スーパーワイドビデオコピー

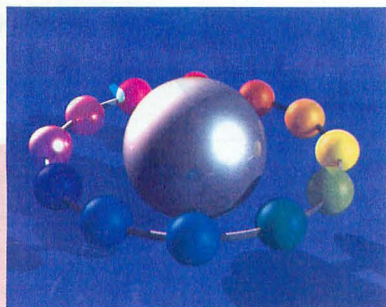
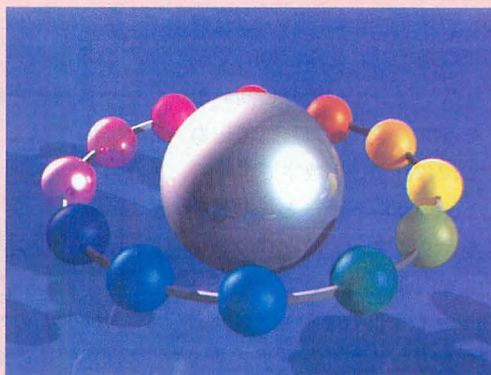
これが今月の特集でご紹介する，1.5倍お得なビデオプリンタによる出力例だ。サンプルはすべて同縮尺（62%）で①がノーマルな大きさ。②は文字領域に同じ画像を加えたもの。見るとわかるように文字領域ではドットの比率がずいぶん違っている。これを補正して縦横を変え，全面にプリントしたのが右側の③と④だ。③は通常の拡大プリント，④は全体の輝度を少し明るめに補正したもの。

①

②

③

④

特集

活用ハードディスク&プリンタ

パソコンは本体だけでも十分にさまざまな可能性を見せてくれる。しかし，用意された周辺機器を接続することにより，思いもしなかったような様変わりを見せてくれる。ハードディスクは記憶装置，すなわちパソコンがすでに持っている機能の一部を強化するものだが，それを接続することにより得られるのは，コンピュータ環境の量的な変化ではなくて，むしろ質的な変化である。
ひところのフロッピーディスクドライブがかなり安くなっても1ドライブ5万円程度（5インチ2Dで）だったことから考えると，40Mバイト10万円というハードディスクユニットの市場価格は驚異的ですらある。
ハードディスクの場合，やっと気軽に使える時代にはなったが，X68000ではとりあえず接続するということだけでも大問題となる。現在はまだ「なにができるか」といったことを考える段階ではなく，それに至る前段階を整理しておかなければならない。
そしてプリンタは出力装置としてパソコン単体では絶対にできないことを実現してくれる周辺機器だ。多くのプログラムで処理の最終目的となることもあり，パソコンの一部ではなく，対等な立場からCPUをアゴで使うことのできる周辺機器のひとつともいえるだろう。
1988年11月号のプリンタ特集で運用のためのひととおりの解説を行っているので，今回はもう少し違った角度から活用を試みる。すなわち，プリンタをシステムとして考えたときに「なにか足りないものはないだろうか？」「もっと自由に使えないだろうか？」という希求に対する解答例だ。これらは，またさらに新しいシステムへの可能性を見せているようにも思われる。引き続きあなた自身で次なる解答を導き出してみてほしい。

CONTENTS

賢いハードディスクの選び方

7機種接続&総チェック

Nakano Shuichi
中野 修一

まずはどういったハードディスクが接続できるのかというところから見ていきます。各社から集めた7種類のユニットをX68000に接続し試用した結果とハードディスク選択時の注意点をまとめてみました。

5月号の使用機種集計で見るとOh!X読者のハードディスク所有率が異様に高いことがわかる。サンプル数1000（無論無作為抽出）に対して，242台という数字ですから，これがほとんどX68000ユーザーと考えるなら実に57%にもなります。このうちのかなりの数はオマケで内蔵型のついてきた(?)ACE-HDユーザーかもしれません。多少データの偏りがあるとしても，ハードディスクに関心を持っている人は少なくないと思います。

今回はハードディスクの導入と活用をテーマに特集を組んでみました。

それではまずは導入編からいってみましょう。

なにが使えるのか

X68000のハードディスクインタフェイスはSCSIの縮小版といった半ば独自のものを採用しています（イニシエータにしかならない）。しかし，ハードディスクをつなぐ際には大は小を兼ねるという意味から，SASI（PC-9801などのインタフェイス）の拡張版と考えるとよいでしょう。

SASIという規格は統一規格としてはSCSIに及びませんが，IBM-PCに採用されたことにより現在までのところ実質的な共通規格になっています。あまりハード的な制限によらず，論理的なブロック番号とコマンドでデータの受け渡しを行っているため，ふつうにつなげばたいていつながる「はず」です。

しかし，接続してもうんともすんともいわないもの，つなごうとしてもコネクタが違ったりする場合さえあります（メーカーによっては変換ケーブルを用意しているところ（ICM，緑電子など）もあるので店頭で見てコネクタが違うものは変換ケーブルの有無を問い合わせてみるのがよいでしょう）。

今回，ハードディスクメーカーの協力でX68000にPC-9801用に市販されているハー

ハードディスク紹介

Logitec LHD-34V

☎03(257)6371

Logitecはシャープにも OEMとしてハードディスクを供給している実績のあるメーカー。純正の外付け40Mバイトハードディスクは供給されていないが，さすがにX68000との相性はよいようだ。オートリトラクトも備え，アクセス速度ほか性能的にもまったく申し分ない。

このハードディスクユニットはインタフェイスにF-1チップを搭載しており，PC-9801シリーズで使用する場合には2.7倍の高速転送が行えるが，専用インタフェイスのないX68000ではとりあえず関係ない。前面にこのチップを使用するかどうかを選択するスイッチがついており，一応低速モードで使用すれば問題はないはずだが，実際は高速モード・低速モードどちらでも動作するようだ。ただし，速度は変わらない。

高速モードが使えないなら旧型でもあまり変わりはない。ただし，編集室で使用しているLHD-34HR（34Vのひとつ前の型）ではなぜか20を超えるパーティションがうまく切れない。それ以外は40Mバイトをフルに使うこともできれば，小さなパーティションで区切ることもできる。OS-9でもまったく問題はない。

itec ITE-403

☎06(532)0120

X68000専用として発売されている数少ないハードディスクのひとつ。容量は40Mバイト，実売価格も手頃で，平均アクセスタイムもこのクラス最高レベルで，と非常にコストパフォーマンスが高く，特に難点のないハードディスクユニットであるといえる。

X68000専用というだけあって，HumanでもOS-9でもなんでも動作する(当たり前か)。本体にあわせてグレイとブラックの2色が用意されているのもうれしい。最近のハードディスクはPC-9801専用として機能アップされていることが多く，その弊害として他機種での使用時に障害となることも多い。X68000で使う分には，分割使用を想定したり特殊コントローラへの対応などの余計なことがしてないほうがいいのだ。

また，オートリトラクト機構を採用しており，電源電圧降下が起こるとヘッドをシッピングゾーンに待避する。ただし，パソコンからctrl-Cを送ってもシッピングは行われない。シッピング時にヘッドを待避する音が「カシャン」と響くのはいまいち気になる。ヘッドを機械的に固定しているためと思われるが，ほかのオートリトラクト機ではこんな音はしないのだが。

ドディスクユニットを接続試験できることになりました。今回取り上げたものについては一部制限つきながら，なんとかX68000に接続が可能です。

それでは，それぞれのハードディスク紹介を交えながら，X68000用ハードディスクの選び方を考えてみましょう。

容量は適切か

5Mバイトで始まり，10Mバイト，20Mバイトと容量の増えてきたハードディスクですが，最近は40Mバイトが標準といっていいでしょう。といっても，ハードディスクの容量に特に基準はありません。OSのサポートしている範囲なら大きければ大きいほどいいのは当然ですね。

ふつうのHumanが一度に管理できるのは40Mバイトまでです。しかし，ハードディスクによっては1台を分割して使用することもできますから，40Mバイト以上のハードディスクでも簡単にあきらめてはいけません。今回は80Mバイト，140Mバイトのドライブも扱ってみました。さらに6月号で紹介したハドソンの384Mバイトのものもありますが，個人で使える値段ではないのでここでははずしておきました。必要ならば1989年6月号を参照してください。

選択時の問題は予算との兼ね合いに尽きます。ハードディスクは容量が大きければ速く，値段も高くなるものですから，かなり迷ってしまうところでしょう。現状では標準的な40Mバイトのものを選ぶのが無難なところです。ふつうの人には特に高いものは必要ありません。

しかし，接続できそうな安くて遅くて小さなハードディスクがあった場合なら少し考えてもよいかもしれません。容量が小さくても，あるとないとではコンピュータの環境がまったく変わってしまいます。最初のうちは容量を気にすることもないでしょう。

しかし，使い込まれ，残り容量が2Mバイトを割ったハードディスクというのはフロッピーディスクよりも悲惨なアクセス速度となります。10Mバイトはいつでも使えるというくらいの余裕を持った使い方がハードディスクを快適に使うコツです。容量が少なくなったらバックアップをとり，中身を整理します。このときハードディスクをフォーマットしなおしてファイルを順に入れなおせば速度が復活します。

速度は十分か

ハードディスクは速いほうがいいに決まっています。速度は平均アクセス速度，転送速度といったもので表されます。平均アクセス速度は一般的に見て小型ディスクのほうが速くなります。これはヘッドの移動

ハードディスク紹介

緑電子 DAX-H5V

☎045(961)7511

横置きタイプの40Mバイトハードディスクユニット。特に問題点はなく，HumanでもOS-9でも自由に組み込める。拡張用ドライブの機能も持ち，装置番号は0～9に変更可能。

最近のものに比べてややアクセス速度が遅いのが少し気になるが，それ以外にはまったく問題点はない。平均アクセス速度は性能表では40ms，マニュアルの冒頭では65msとなっている（表1では性能表のほうを参考にした)。また，性能とは関係ないけど，明るい緑と赤のLEDがとっても綺麗。

サービスコンセントをひとつ持っているがこれは増設用のDAX-H6専用のもの。そのほかの用途での使用には動作保証されていない。

横幅はX68000（PRO以外）とほぼ同じ。このハードディスクの上にX68000本体を置き，隣に21型ディスプレイを並べるとちょうどいい高さになる。このようなレイアウトは多少乱暴だが，このハードディスクはもともと同型のハードディスクユニットを上に積み上げるように設計されているので，電磁気などの問題はあまりないと思われる。省スペースというのも外付けハードディスクの重要なポイントだ。

ランドコンピュータ LDM-540

☎06(304)8424

いかにもヘビーデューティといった感じの40Mバイトハードディスクユニット。平均アクセス40msでオートリトラクト機構を備える。

オートリトラクトといっても，一般的な電圧降下によるヘッド待避機構ではなく，2.5秒以上アクセスがないと自動的にヘッドをシッピングゾーンに待避するという設計になっている。不意の地震などでも比較的安全に作業ができるということだろうか。電源を切る場合には最終アクセスから2.5秒以上間をおくことが必要（X68000からctrl-Cでシッピングすることも可能)。実際に使ってみると，なんらかのコマンドを使ったとき，そのコマンドが起動したあと，少し間をおいてもう一度ディスクアクセスが起きる。2.5秒といってもほとんど直後という感じだ。慣れないとなんとなく不自然に感じるかもしれない。

面白いのはライトプロテクト機能を持つということ。ディスク全体にしか指定できないが，電源スイッチとなるキーをW.P.の位置にすることでハードディスクへの書き込みをハード的に禁止できる。開発中のプログラムの暴走でデータを失ったり人為的なデータ破壊を未然に防ぐことが可能だ。

量が違うため起こる物理的な傾向です。

平均アクセス速度が28msだと確かに使っていてもレスポンスが速いように感じますが，具体的にどの程度違うのかははっきりしません。ACE-HDの内蔵型が80ms，では最新の28msドライブが3倍速いかというとそうでもないようです。

また，ハードディスクの速度はドライブユニットの基本性能だけでなく，コントローラの性能によって変わってきます。

ものによっては，一度読み出したデータをキャッシュメモリという内蔵RAMに保存し，書き込みなしで同じ領域を呼び出されたら，ディスクをアクセスせずにこのキャッシュメモリから呼び出して高速転送するという機構を備えています。実際，ディスクを使用する場合，同じ領域を何度もアクセスすることが多いのです。このようなキャッシュ搭載機は同じスペックの機械よりかなり高速に感じられます。

また，最近のPC-9801専用のハードディスクには高速化のため専用のインタフェイスでしか動作しないものもあるようですので注意しましょう。

その他の注意

そのほか，ハードディスクを導入する際に考慮すべき点をまとめてみると，

●OS-9での使用ができるか

OS-9ではディスクの管理がHumanとは違いますから，Humanで動いてもOS-9では動かないということがあります。今回試験したドライブでもいくつかのものはOS-9でのフォーマットができませんでした。Humanに比べフロッピーディスクのアクセスは遅いので，OS-9である程度のことをしようとするとハードディスクは必需品といえますから，これは残念です。

ただし，OS-9を使わずHumanのみで使う分にはまったく問題ありませんから，これだけでそのディスクが使えないと考えるのは早計です。

●X68000本体のバージョンによる障害？

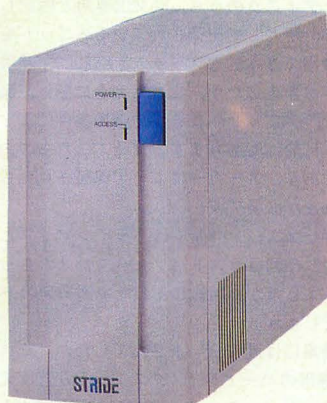
ICM製のドライブユニットでときどき発生します。ディスクアクセスの途中でエラーが発生しプログラムの実行が停止するのです。基本的にハードディスクまわりの回路は変更されていないのですが，ACE,PRO,EXPERTでは大丈夫なのに初代X68000では少し問題があるようです。

●分割使用について

一部の機種には40Mバイトの領域を20Mバイト2つに分割できるものがありますが，これはMS-DOS上のアプリケーションのなかにはどうしても2台のハードディスクを必要とするものや20Mバイト以上の容量ではうまく動作しないものがあるためつけられたモードです。Humanで使っている限り，あまり関係がありません。この機能を使っ

ハードディスク紹介

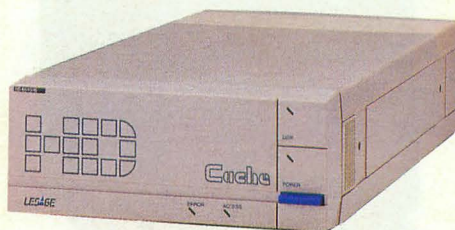
ICM STRIDE SR-80

☎06(647)2008

80Mバイトの容量を持つ大容量ハードディスク。ただし，通常のインタフェイスを使う限りでは40Mバイト2基としてしか使用できない（というよりFORMATコマンドの問題か？）。しかし，実質的には40バイトを超えるファイルを扱わない限り40×2でも，使い勝手などで問題になることはないだろう。

通常，ハードディスクの転送速度は5Mbit/sなのだが，この機種では10Mbit/sの転送速度を実現している。そのためか，40Mバイトの領域確保などを行っても他機種より数段速い。アクセス速度もトップクラス，本体内にキャッシュメモリ（40Kバイト）を内蔵し，さらに高速化を目指している（キャッシュの使用に特殊な操作は不要）。

難点はOS-9でFORMATできないこと。さらに旧型機種では動作が不安定なこと。初期型のX68000以外でHumanを使用するなら問題はない。また，ICMのハードディスクユニットはハードディスク側のコネクタに特殊なものを使用しているのでX68000と接続するときには，変換用のケーブル（HD-3020-01：税別7,500円）が必要となる。購入時にメーカーに問い合わせてみてほしい。

ICM LESAGE 6040HS

☎06(647)2008

横置き40Mバイトタイプのハードディスク。この機種も本体内にキャッシュメモリを内蔵している。なんと標準で1Mバイト，最大6Mバイトまで拡張が可能だ。これらのキャッシュはごくふつうにハードディスクをアクセスするだけで内蔵のコントローラが勝手にキャッシュを見にいってくれるので使う側は余計なことをなにも考える必要はない。転送速度はごく一般的な5Mbit/sながら，平均アクセス速度は28msと高速。大容量のキャッシュを考えあわせるとクラス最高の高速性を持っているといえよう。

注意点は前述のSTRIDEと変わらない。OS-9での使用と旧機種での使用はあきらめたほうがいい。また，これも専用のコネタタを採用しているのでX68000との接続には別売りの専用ケーブル(STRIDEと同じもの）が必要となる。

そのほか，ICMのハードディスクマニュアルは非常に力が入っている。大半はX68000ユーザーには不要なマニュアル(MS-DOSでの注意など）なのだが，ハードウェアマニュアルだけでも94ページのボリュームがあり，内容もわかりやすい。使用機種と用途が限定されるのが残念だ。

て分割したディスクは単にパーティションを切った領域と違い，別々の装置番号が割り当てられます。

導入編のまとめ

今回テストしたハードディスクでX68000に接続できなかったものは，キャラベルデータシステムのCA-40LG(24ページのカコミ参照)，緑電子のLittle Bシリーズおよび POKEDYシリーズ，コンピュータリサーチのCRC-MH4など。このうちキャラベルのものは最新の3.5インチ版ですが，旧型の5インチ版はユーザーの栗野雅彦氏によってHuman，OS-9など問題なく動作することが確認されています(H540S)。

そして，今回は特に取り上げませんでしたが，純正ドライブ（内蔵型）はもちろんなにも問題なく動作します。PRO，EXPERT にしか使用できませんが，コストパフォーマンスはそう悪くありません。欠点としてはスイッチが切れないことくらいでしょうか。なんといっても絶対的な省スペースが可能になります。PRO,EXPERTユーザーの方は当然第一候補に挙げておくべきでしょう。

＊　＊　＊

今回試用したのは山ほどあるハードディスクのほんの一部にすぎません。すでに他社のハードディスクを使用している方はぜひ編集室まで情報提供をお願いします。

続いて活用編です。導入したハードディスクの有効利用を考えてみましょう。

RAMディスクの有効活用

まず，X68000本体を1Mバイトメインメモリのみで使っている方はハードディスク導入より先に2Mバイトに増設することをおすすめします。1MバイトのRAMがあれば，ほとんどのアプリケーションは走ります。しかし，使用環境を整備しようとするとどうしてもRAMが足りません。特にハードディスクを接続するとなにかとメモリを消費してしまいます。

さて，ハードディスクの普及とともに編集室では辞書をRAMディスクに載せるのをやめる人が増えています。ではどうしているかというとハードディスクに辞書用のパーティションを設定してやるのです。

RAMディスクは200〜300Kバイトにとどめ，通常の作業はほとんどRAMディスク上で行います。ハードディスクでやるべきこととRAMディスクでやるべきことをしっかり把握しておけば，どちらも最大限の力を発揮できます。

ハードディスクにいつなにが書き込まれてもかまわない領域を設定するなどすれば，RAMディスクは思い切って小さめにできるものです。その空いた分をメインメモリ

ハードディスク紹介

CTS HDα-100

☎03(835)1887

140Mバイトで20万円台という非常にコストパフォーマンスの高い大容量ハードディスクユニット。接続にはハドソンから出ているSCSIボード，動作にはHuman68k ver2.0以上および，SCSI用デバイスドライバが必要。こちらについてはシステムショップハドソンに問い合わせてみてほしい。まず，BINDコマンドでFORMAT.Xを拡張してSCSIデバイス対応のものを作成しておき，フォーマット時はハドソン製の384Mバイトの大容量ディスクを指定して140Mバイトの領域を確保する(マニュアル上の容量と少し違うので注意)。SCSI用のデバイスドライバが必要なので，当然(現在のところ)OS-9には対応できない。

本体は3つのブロックに分かれ，ディスクユニット，電源，コントローラからなる。将来的にはストリーマなども内蔵できるように設計されている。その分，筐体は大きくなってしまっている。

この140Mバイトのタイプ(ほかに80，300，600のシリーズがある)はHDαシリーズのなかでもっとも遅いものだが，それでも速度的にはほかのハードディスクより優位にある。従来のパソコン用とは一線を画す，本格的なハードディスクユニットだ。

AFTER REPORT OF THE TEST

今回テストしたディスクのうち，X68000専用と明記してあるもの以外は，すべてPC-9801用として出荷されているユニットを使用しています。表1にこれら7機種の基本的なスペックとX68000で使用する際の基本的な注意点を挙げておきましたが，これも完全に動作が保証されたものではありませんので，アプリケーションや使用条件によってはなんらかの問題が生じてくる可能性もあります。また，これらの内容は本誌が独自に調査したものであり，メーカーによって保証されたものではありません。くれぐれも注意が必要です。

ここで取り上げたユニットの多くは，ラインアップとして別容量の製品を持っています。今回はコストパフォーマンスのもっとも高いと思われるものを紹介しましたが，用途にあわせて同系列のドライブを検討してみるのもよいでしょう。

なお，今回の特集には間に合いませんでしたが，これら以外にもアイテムからX68000専用のハードディスクユニットが近々発売予定となっています。これでサードパーティによるX68000専用機としては4台目になります。やはり，メーカーによって動作が保証されているというのは大事なことですから，こういった動きはおおいに歓迎すべきですね。

また，今回接続できなかったキャラベルのCA-40LGはX68000用に特殊フォーマットを施したものも用意されており，それを使うと Human68kでなら問題なく動作するようです。ただし，特殊なフォーマットのため，ユーザーによる再フォーマットができませんので注意が必要です（よってOS-9不可)。

今回紹介したハードディスクの多くがPC-9801の横に置いて使うことを前提にされています。X68000の横に並べるには，デザイン的にも多少の工夫がほしいところです。X68000のマンハッタンシェイプは本体とディスプレイだけで完結しており，もはや横にハンパなものを置いても絵になりません。ハドソン製の384Mバイトハードディスクなどは，ドライブ自体は筐体の下半分で収まるのに，わざわざ拡張I/Oボックスと大きさを揃えていましたしね。専用機が出てくるのなら，性能だけでなくデザインにも気を配ってほしいものです。

ハードディスクの普及率があがれば，それに対応したアプリケーションなども期待できます。この記事が皆さんのハードディスク活用の手助けになれば幸いです。

に当ててやれば，かなり自由で快適な環境が構成できます。

ディスククラッシュについて

フロッピーディスクでは磁気ヘッドと磁性面が直接接触して信号の読み書きを行いますが，フロッピーディスクの10倍以上の回転速度を持つハードディスクでこのようなことを行うとディスクが傷だらけになってしまいます。ハードディスクではヘッドは磁性面から浮いており，常に非接触でアクセスを行います。ですから，通常の使用状態では，ハードディスクはフロッピーディスクよりもはるかに安全な記憶装置なのです。

しかし，浮いているとはいえ，ヘッドと磁性面の間はミクロン単位でしかありません。衝撃が加わるとヘッドは磁性面に接触し，データを破壊してしまいます。

ハードディスク関係の本を読むと，回転が止まり空気の流れがなくなるとヘッドが磁性面に密着してしまい，データを破壊するので必ずハードディスクの使用後にはヘッドをシッピングゾーンというところに待避することが必要だと書いてあります。

ふつうは衝撃を与えない限り，ヘッドを待避しなくてもそれほど問題はないはずなのですが，昔はディスクの電源を切るたびに確実に1クラスタずつディスクを破壊していき，ある日突然起動不能になって使用者を驚かせるというロシアンルーレットのような機種がありました。

最近はさすがにこういったことはなくなりました。X68000では本体の電源OFF時にハードディスクのシッピングを行うので，本体，ハードディスクの順に電源を落とせばまず間違いはありません。

さらに電源を切ったり，停電などで電圧が下がったりすると，電源が切れる前に最後の力を振り絞って自動的にヘッドを最内周に待避するというオートリトラクト機能を持った製品も増えています。

ではシッピングしておけばハードディスクは安全かというと，これも考え方しだいといったところです。ICMのマニュアルから引用してみましょう。

「また，『シッピングゾーンに移動させる』という考え方についても弊社はあまり積極的に推奨はしません。ストップキーを押すことによって，ヘッドがシッピングゾーンに待避していても，強い衝撃を与えればヘッドが磁気媒体に衝突し『ダスト』と呼ばれる磁気粉末をエンクロージャ内に飛び散らせます。そして，このダストは確実にハードディスクの性能劣化につながり，故障の発生確率を高めます。シッピングゾーンにそのたびごとに移動させなくても，また，少々の衝撃が与えられても障害を起こさないハードディスクユニットの実現が根本で

表1　各社ハードディスク一覧

メーカー	Logitec	itec	緑電子	ランドコンピュータ	ICM	ICM	CTS
型番	LHD-34V	ITX-403	DAX-H5V	LDM-540	STRIDE SR-80	LESAGE 6040HS	HDα-100
価格	158,000円	188,000円	198,000円	298,000円	198,000円	290,000円	245,000円
容量	40Mbyte	40Mbyte	40Mbyte	40Mbyte	80Mbyte	40Mbyte	140Mbyte
平均アクセス時間	28ms	28ms	40ms	40ms	28ms	28ms	23ms
データ転送速度	5Mbit/s	5Mbit/s	5Mbit/s	5Mbit/s	10Mbit/s	5Mbit/s	5Mbit/s
キャッシュメモリ容量	なし	なし	なし	なし	40KB	1MB(〜6MB)	なし
横幅	95mm	98mm	155mm	140mm	88mm	200mm	167mm
奥行き	145mm	147mm	280mm	330mm	280mm	375mm	356mm
高さ	290mm	340mm	81mm	180mm	150mm	83mm	210mm
重量	3.5kg	3.4kg	3.5kg	6kg	3.2kg	5.4kg	10kg
消費電力	24W	28W	＜40W	＜70W	35W	45W	47W
AC OUTLET	—	—	1	2	—	—	—
旧機種	○	○	○	○	×	×	○
OS-9	○	○	○	×	×	×	×
ID変更	0, 1	×	0〜9	0, 1	0, 1	×	×
オートリトラクト	○	○	×	○	×	×	○
マニュアルページ数	93P	22P	16＋32＋22P	64P	112＋390＋260＋6P	94＋420＋260P	44P
インタフェイス	標準	標準	標準	標準	標準	標準	SCSI
拡張	○	×	○	○	○	○	×

あると考えています」

と，このように書いてあっても，同じマニュアルに電源OFF時にはヘッドをシッピングしてくださいと書いてあるのですから，やはりシッピングをしないよりはしたほうがいいのでしょう。ふつうのハードディスクではシッピングエリアはヘッドの耐震性を増すような設計になっているようです。

HumanV2.0での注意

もともとハードディスクは規定の容量以上の容量を持っています。また，出荷時すでにいくつかの欠陥トラックを持っていることも考えられます。規定以上の部分は，これらの予備として使われるように設定された部分なのです。

ですから，少々欠陥トラックがあっても使うたびに増えたりしない限り欠陥品ではありません。OSもそれを考慮して欠陥トラックがあった場合，フォーマット時に代替トラックという部分で代用するように領域を確保していきます。これで見掛け上ちゃんと欠陥トラックのないディスクのように動作します。

ところがHumanVer2.0のFORMATコマンドでは，たまにこの代替トラックの割り当てに失敗することがあります。使っていてどうしてもおかしいようなら，一度Ver1.0でフォーマットしなおしてみるとよいでしょう。

最後にDIRコマンドの高速化について。20Mバイトではあまり気にならないのですが，Humanでハードディスクのディレクトリをとる場合，たとえばEXPERT-HDの40Mバイトディスクでは DIRコマンド実行後，カーソル点滅3回分の間をおいてディレクトリが表示されます。フロッピーディスクなら瞬時に表示していたものがなぜか遅くなっています。

これはディレクトリ表示の際にディスクの空きエリアなどを表示しているためです。せっかく高速HDDをつけ，TurboConsoleなどで表示を高速化しても，これでは悲しくなってしまいますね。

そこで高速DIR用の変更をやってみました。これはシステムの書き換えですので，あくまでも自分の責任においてやってください。command.xVer2.0の場合，3CAD$_H$バイト目からのFF36を7000に書き換えます。バージョンによって違いがあるかもしれないので，必ず確認すること。

これでディレクトリがノーウェイトで表示されるようになりました。ただし，ディスクの使用量と残り容量の表示が0になってしまうので注意。

まとめにかえて

Cコンパイラ，Z'sSTAFF，PrintShopなど，大きなアプリケーションではハードディスクを使わないと使いづらいこともあります。最近はハードディスク対応のゲームまで作られるようになり，ハードディスクの用途も広がっています。

ハードディスクを導入するということは，自分自身の環境を作るということです。これは，単にちょっと大きな外部記憶をつけたというだけではありません。かといって，ものすごく大きな主記憶を得たわけでもありません。使い方しだいで非常に便利なものにもなりますし，使い方を誤ると逆にうっとうしい存在になるかもしれません。

思考錯誤を繰り返してシステムを構築していくことは実はとても楽しいことです。バックアップをとり，中身を整理していくたびに，使いやすいシステムになっていくのがよくわかります。これもパソコンの楽しみ方のひとつかもしれませんね。それでは，ハードディスクによる新環境をいろいろと楽しんでみてください。

基礎から学べる

ハードディスク雑学講座

HARD DISK

Kuwano Masahiko
桒野 雅彦

最近，注目の度合が増してきたハードディスク。しかし，その箱の中身の実態までは，そう簡単に知ることはできません。そこで，基本的構造から仕様解説と，あれやこれやを詰め込んで徹底解説をお届けすることにしましょう。

第1章 旅立ち

四角い箱の中身

外部記憶装置というのは，デジタル信号だらけでいったいなにをしているかわからないコンピュータのなかでは，珍しく物理的な動きを伴う装置です。それだけに，一般の人のコンピュータというもののイメージの多くは，外部記憶装置といったものであったようです。昔の映画やマンガなどでは，吐き出された紙テープを両手で広げて「大変だ……」なんて台詞をしゃべっている場面があったりして，四角い鉄の箱のなかで，磁気テープがグルグルと回っているのがコンピュータでございとばかりに大きな顔をしています。

紙テープなどはいまだに大学の実習などで使われることもあるようですが，さすがに人間がノリとハサミで修正できるような超低記録密度というのもさることながら，ポテトチップを1週間も出しっぱなしにしておいても美味しく食べられるような環境は望むべくもない日本では，メディア自体の伸び縮みも問題で，雨の日用のテープなんていうものが必要だったりと，使い勝手の面ではあまり関わりたくないしろものでした。

一方，磁気記録のほうは確実に発展を遂げています。大容量を誇る磁気テープは，ハードディスクのバックアップ用としてはいまでもしっかり使われています。この磁気テープは，容量は大きいものの頭からテープを巻き取りながら読むしかないので，目的のデータにありつけるまでの時間がかかります。これを改善したのが磁気ドラムで，円筒を回転させながら，その表面に記録していく形式のものです。円筒を一定の回転数で回しておいて，ヘッドを回転軸と平行に移動すればよいというわけです。

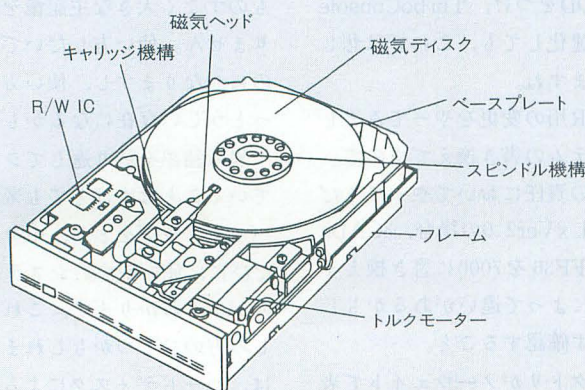
磁気ドラムは円筒の表面しか利用できないので，占有する空間の割には容量が上げられません。半径を2倍にすると円周は2倍になりますから容量も2倍弱くらいにはできるのですが，逆に底面積は4倍にもなってしまいます。それに，ドラムのなかはまったく無駄な空間となってしまいます。

これをレコードのように円盤(ディスク)の表面に記録するようにすれば，空間が有効に利用できるではないかというのが磁気ディスク装置の考え方です。磁気ディスクは，ヘッドとディスクを分離したものと，一体にしたものに分けられます。

前者の代表はフロッピーディスクでしょう。以前は大きなディスクパックを使ったハードディスクもありましたが，いまではまず見掛けることはありません。直径が60センチもあるくせに10メガくらいしか入らず，電源を入れるとしばらくフィルタを通した空気を循環させてからおもむろに動き出すというやっかいなしろもので，デカイ，重い，うるさい，オマケに大飯（電気）喰らいでありました。光磁気ディスクも前者のタイプですが，こちらは小さい，静か，軽い，少食，そのうえ容量はハードディスク10台分と，技術の進歩を見せつけてくれています。

一方，後者の代表はなんといってもヘッドとディスクを一緒にして密封した（俗にウインチェスターと呼ばれることもあった）ハードディスクです。ウインチェスターというのは，IBMの開発コードで，それが一般的に用いられるようになっていましたが，最近ではハードディスクといえば密閉型が当たり前になったせいか，あまり耳にしなくなりました。

ハードディスクの場合，磁気ドラムやフロッピーに比べてかなり高い記録密度で，しかもディスクの回転数も約10倍と大きくなってくると，ヘッドとディスクの位置関係を正確に保つ必要があり，またちょっとしたゴミの付着もエラーのもとになってきます。ですから，ヘッドとディスクを一緒にしてクリーンルームのなかで組み立て，密閉して作られます。

図1 ハードディスクの内部構造

黒船の来日騒ぎ

さて，パソコンの外部記憶装置のことを思い出してみることにしましょう。個人のホビーとしての小型コンピュータでは，カセットテープが外部記憶装置のスタンダードとして長く利用されてきましたが，OA化の波が押し寄せるにつれて，さすがにカセットでは厳しいものがあるということで，次第にフロッピーディスクが使われるようになり，ドライブ内蔵機種が増えたこともあってここ5年ほどの間に一気に普及しました。

今日のパソコンのような量産機ともなると，コストに対する要求が厳しく，ドライブメーカー間の競争の激化とともに，いまではハードウェアメーカーが仕入れている値段は1台数千円ということです。ドライブのメーカーのほうもかなり苦しい商売を強いられているようで，撤退の噂さえ流れているメーカーも2，3あるようです。

フロッピーディスクが安くなれば，次はハードディスクの番だというわけでもないでしょうが，あれだけ高かったハードディスクも，ここ1，2年ほどの間に一気に安くなり，いまでは40Mバイトでも売値は10万円を切るのが当たり前のようになっています。

このきっかけになったのはおそらく，アメリカのシーゲート社のドライブを使って10万円ちょっとでPC-9801用のハードディスクが売り出されたことでしょう。その後，モデルチェンジをして，前のモデルを半額以下で売り出すところが続出したところをみると，高値安定市場が突如，黒船騒ぎを起こしたといったところでしょうか。

アプリケーションも大きくなり，いろいろなものを連係したりするようになってきて，ハードディスクが欲しくなってきたところに，かついちばん値が張る周辺機器であったものが，プリンタと同じくらいのところまできてしまったのですから，普及にも弾みがつこうというものです。かくして，ショップの周辺機器の棚はプリンタとハードディスクが占拠するようになったわけなのです。

第2章
ハードディスクの構造

まずは基本構造から

フロッピーディスクに限らずハードディスクも，ついに登場の光磁気ディスクも，回転する円盤をまず同心円状の部分（トラック）に分割して，さらにそこをいくつかに区分けしたブロック（セクタ）を最小単位として読み書きを行っています。あるセクタをアクセスするときはまず，ヘッドを目的のセクタのあるトラックの場所まで移動しておいて，あとはディスクが回って，目的のセクタが来るのを待つわけです。いまのフロッピーディスクは両面に記録していますからヘッドは2つ，ハードディスクではディスクを複数枚持つものもあるので，その枚数の2倍の数のヘッドがありますが，これらはすべて一緒にひとつのモーターで動くようになっています。

ヘッドを移動せずにアクセスできるトラックの集まりをシリンダと呼びます。ディスク1枚の両面にトラックがあれば，1シリンダあたり2トラックというわけです（図2）。

余談ですが，CD-ROMの記録方法はちょっと違っていて，データは内側から外側に向かって渦巻き状に記録されています。このためCD-ROMでは任意のセクタを読み出すランダムアクセスが苦手なのです。

大容量の秘密

記録のやり方が同じなのに，ハードディスクがフロッピーよりも圧倒的に大容量であり，かつ高速であるというのはどこに違いがあるのでしょうか。

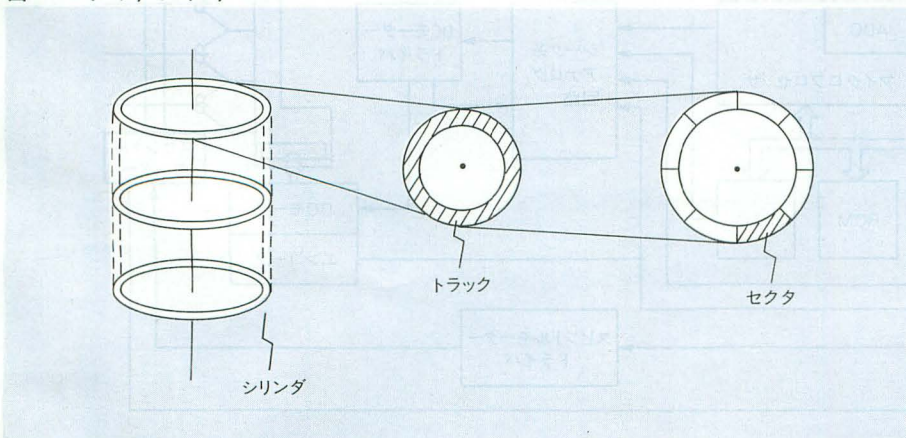
まず，大容量ということから見てみましょう。ハードディスクはフロッピーよりも高密度に記録してあるとはよくいわれますが，考えてみれば，同じ大きさで容量が大きいなら高密度であるのは当たり前の話でしょう。そこでEXEクラブ会員証兼電卓を片手にデータブックを調べてみました。すると意外なことがわかりました(表1)。ハードディスクの記録容量はフロッピーよりも圧倒的に大きくなっていますが，個人が手にできるようなものに限ってみると1トラック当たりの容量は確かに増えてはいますが，フロッピーの場合とそう大きくは(何10倍も）違っていないのです。

たとえば，NECの5インチのハードディスクでPC-9801用の純正ドライブといわれている，D5146/5126の1トラック当たりの容量は8Kバイトですが，これはフロッピ

表1：D5146Hの基本仕様

項目	内容
記憶容量	
アンフォーマット時	
1ドライブ当たり	51.24Mバイト
1シリンダ当たり	83328バイト
1トラック当たり	10416バイト
フォーマット時	
1ドライブ当たり	40.3Mバイト
1シリンダ当たり	65536バイト
1トラック当たり	8192バイト
ディスク枚数	4
ヘッド数	8
シリンダ数	615
ディスク回転数	3600±0.5%
トラック密度	700TPI

図2　シリンダ，トラック，セクタ

ーの2HDフォーマットのときと同じです。容量を圧倒的に違うものにしているのは，トラックの密度によるものでした。1トラック当たりの容量はあまり変わらなくても，トラックの数自体が圧倒的に多いので，大容量になっているのです。

トラック密度の単位はトラック/インチ(TPI)，つまり半径方向の1インチ当たりに何本のトラックがあるかという数値が一般に用いられます。5インチのフロッピーではせいぜい60TPI程度ですが，先ほどのD5146/5126では，700TPIもあります。5インチのフロッピーではディスク片面のトラック数は約80トラックありますが，D5146/5126では615もあります。トラック密度はディスクの小型化とともに増える傾向にあり，3.5インチの40Mのものでは1000TPIくらい，さらに最近開発の進んでいる2.5インチのものでは1200から1500TPI程度まで上げて，大容量化が図られています。

高速化の謎

次に高速であるという面について見ていきましょう。高速にするということは，目的のセクタからいかに早くデータを引き出すかということです。この時間はディスクから引き出す準備が整うまでの時間と，データの転送を開始してから終了するまでの時間に分けて考えることができます。前者のデータ読み取り準備のための時間のことをアクセス時間といい，後者のデータを取り出す速さをデータ転送速度といいます。

この両者に共通するファクター(要因)のひとつがディスクの回転速度です。ディスクの回転が速ければ，たとえ目的のセクタがヘッドの下を通りすぎたあとであっても，すぐに一回転してきますし，セクタのデータを読み出す時間も少なくてすむはずです。

調べてみると，なるほど大きく違っています。2HDのフロッピーディスクでは1分間に360度回転が標準的なところですが，ハードディスクのほうはというと，遅いものでも2900回転，D5146も含め大半が3600回転程度と，フロッピーディスクの10倍もあります。

ヘッドの移動も，大半のフロッピーディスクではステッピングモーターを使っており，隣のトラックに移るまでの数ミリ秒×移動するトラック分だけ時間がかかりますが，ハードディスクではDCブラシレスモーターや，ボイスコイルモーターといった，連続移動に有利なものを使っています。

ステッピングモーターは，正確にある角度だけ回転させるというのは簡単にこなすのですが，それを除くとステッピングモーターはトルク(回転力)はない，電気は食うわ，回転数は上がらないわで，まるで良いところがありません。

一方のボイスコイルモーターなどはこれと正反対です。正確な回転角を得ることさえできれば圧倒的な高速化が図れます。このようにハードディスクでは，速度重視でモーターが選択されています。懸案の位置制御のほうは，回路の集積技術にものをいわせて，手馴づけています。

まずホール素子やエンコーダを使ってヘッドの位置センスを行います。この情報からヘッドの現在の位置と目標位置の差，現在のヘッドの移動速度などがわかります。これらのデータによってモーターに流す電流にフィードバックをかけ，さらに位置校正信号などによる補正を加えることで位置決め精度の確保を図っています。

D5146のあとに登場したD5146H(登場したのは3年くらい前)では，ヘッド移動にはボイスコイルモーターを使い，ヘッド位置検出はエンコーダ，位置校正にはデータ面のサーボ情報をA/Dコンバータで読み取ることで行って，これらの制御のためにドライブ自体にマイクロプロセッサを内蔵させています。この方法で，シーク速度の最高値は4倍程度まで上がったそうです。

図3　D5146H装置回路ブロック図

そのD5146Hでも，平均アクセス時間は40ミリ秒で，これは個人向けとしてもいまでは決して速い部類ではなくなってしまったのですから，ここ2，3年の進歩はやはりたいへんなものだと感じます。

第3章 フォーマット容量

消えた容量の行方

フォーマット容量と，アンフォーマット容量の差というのはフロッピーディスクでも経験することです。たとえば2HDのフロッピーではアンフォーマット容量は1.6Mバイトですが，実際に使えるのは1Mバイト程度です。

フォーマット容量はFORMATコマンドなどを実行し，初期化処理を終わったときの記憶容量ですから，アンフォーマット容量といえばフォーマットする前の容量のことになります。フォーマットしなければ使えないディスクなのに，アンフォーマット容量とは奇妙な話です。これは次のような理由によるものです。

ディスク上でもデータはすべて1か0かを記録しています。記録用の円盤自身やドライブの仕様などによって，物理的にどのくらいの数の1/0をディスク面に記録できるかが決まってきます。これがアンフォーマット容量です。

一方，ディスクを使う側からすると，この書き込めるデータをすべて自分用のデータ領域として使うことはできません。ディスクを実際に使うためには，ディスクの各トラックをいくつかのセクタと呼ばれるブロックに分割して管理しています。ヘッドを目的のトラックに移動させたあとは，目的のセクタを正しく見つけなくてはなりません。

このやり方には，ディスクの最内周部分に定間隔で穴を開けておいて，この穴に合わせて読み出すハードセクタ方式と，各セクタの先頭に，このセクタが第何シリンダの何番目のヘッドでアクセスされるトラックの第何セクタであるかといったような情報や，読み取りを安定させるための同期パターン領域などをフォーマット作業時に書き込んでおいて，それを手掛かりにアクセスするソフトセクタ方式の2通りがあります。細かい業界内部の動向までは詳しくありませんが，我々が通常手に入れられるもののなかでは，おそらくソフトセクタ方式がほとんどだと思われます。これらの情報のためにユーザーが使える領域が減ってくるわけです。

ハードセクタ方式でもやはりシークにしくじったときに誤ってほかのトラックをアクセスしてはまずいので，各セクタの情報などを書き込んでいます。そのためディスク面をいっぱいに使うというわけにはいかないのです。

第4章 ヘッドと磁気ディスク

かくして電流は流れる

ハードディスクはフロッピーディスクよりも圧倒的に高速，かつ大容量なものですが，基本的な原理は同じで，要は円盤に磁性体をくっ付けたものの上に，ヘッドと呼ばれるコイル（電磁石）を近づけて，円盤を回転させながら電磁石に流す電流を制御してやれば，ディスク上はそれに応じた形で磁化され，電流を流さずとも単に近づけるだけで磁化されたパターンに応じて誘導電流が発生するので，それを増幅してデータとして取り出してやるということです。

誘導電流なんていうと難しく聞こえますが，要するにコイル（単なる電線でもいいわけですが）のそばで磁石が動けば，動いた通りに電流が流れるというだけのことです。

日常的なもので，これを積極的に使ったのにはマイクロフォンなどがあります。別に意識しなくても，電流計やスピーカー，意外なところではモーターに向かって叫んだだけでも結構な誘導電流は発生しているのですから，かなり身近な現象といえるでしょう。

さらにこの応用で，ディスクごっこをしてみるのも面白いかもしれません。記録は少々骨ですが，再生だけなら作るのは簡単です。ヘッドとしてはエレキギターのピックアップが使えるようです。ボール紙などの上に小さい磁石を並べて固定し，レコードプレーヤーや遠心分離器などに乗せてグルグル回しながらアンプをつないだサーチコイルを近づけると妙な音が聞こえてくるでしょう（参考文献：『いきいき物理わく

雑学への基礎知識

ドライブインタフェイス

ドライブ自体のインタフェイスとして多く使われているのにはSMDインタフェイス，ST502/412，ESDIなどいくつかあります。SMDインタフェイスはハードセクタ方式から流れてきたものです。STxxxというのは，アメリカのシーゲート社が自社のハードディスク用に作ったインタフェイスですが，国内外の各社で使われ，標準としての地位を確保しています。これらのインタフェイスはディスクのハードウェアと密着したもので，たとえばヘッドの移動のためにはヘッド移動用のパルスを一発ごとに発生させてやるなど，細々とした部分の面倒をみる必要があります。

X68000などでも使われているSASI準拠のインタフェイスや光磁気ディスクなどで使われたSCSIインタフェイスではこれよりもかなり上位のレベルにあるインタフェイスで，ディスクのすべてのセクタに連番を振って，読み書きなどの処理内容とこのセクタ番号などを与えるだけで処理が完了します。ST502などのインタフェイスとのギャップを吸収するSCSI（SASI）インタフェイスボードというものが長く使われてきましたが，ディスクの小型化，軽量化を求められるなかで別ボードが必要というのは不利なこともあって，いきなりSCSIインタフェイスが出てくるようなディスクも，最近では多くなってきました。

最近登場してきた2.5インチハードディスクである，JD-E2825（日本ビクター）でも，インタフェイスはIBM-PC XT/AT式とSCSIになっています。IBM-PCのほうは世界中で売れてしまった台数からしても，もはや無視できない存在なので仕方ないと考えられるので（IBM-PCの設計のまずさ，非常識さについては機会があれば一度紹介したい），基本的にはSCSIのディスクと考えてよいでしょう。

わく実験』愛知・岐阜物理サークル，新生出版)。

これは目に見える，驚異の低密度記録ですが，体験するには面白い実験といえるでしょう。ここからもっと高密度に記録，再生ができるようにするにはどうしたらよいでしょうか。記録密度を上げるということは，円盤の表面をよりきめ細かく磁化することです。つまり，正確に狙ったポイントを予定通りに磁化することになります。このためにはなにかと広がりそうになる磁力線を極力狭い領域に集めなくてはなりません。また，磁化される領域が狭くなるということは読み出し時の出力の低下を招きますから，強い磁場を作る必要があります。

まず，ヘッドは小さくしなくてはならないでしょう。ヘッドが大きくては，狙った場所以外の場所まで磁化させてしまいます。ヘッドの厚さも，厚くないほうがよいでしょう。磁化する場所が細かくなるほどコイルを近づけたときに得られる出力が減りますから，正確に拾い出すためにもディスク面とヘッドの距離は極力近づけることが必要です。ディスク側でもあまり深いところまで磁化されるようだと，隣のビットの影響を受けやすくなりますから，表面の磁性体の厚さも極力薄くしたほうが有利です。

薄まりゆくヘッドと表層の関係

ディスクのヘッドとして標準的に使われてきたのはフェライトヘッドでしたが，記録密度を上げていくにつれて，Snoekの限界と呼ばれる，透磁率の周波数限界のために10MHzの信号では効率が落ちてくることや，飽和磁束密度が小さい，インダクタンスが大きいために速いデータ転送が難しいなどの欠点が目立つようになりました。最近は薄膜ヘッドやMIG (Metal In Gap) ヘッドと呼ばれるものに移行しています。薄膜ヘッドというのは，薄い導体のパターンを渦巻き状に作り，それを何層も重ねてヘッドにするものです（図4）。

この薄膜ヘッドというのは，紙の上に6Bくらいの濃い鉛筆で渦巻きを描いて，その中心と外側に電池をつないでやると弱いながらも立派なペーパー電磁石になりますが，それをギュッと小さくしたようなものです。当然，製造にはスパッタリングなどのLSI製造技術が生かされています。目的とする形状に精度よくヘッドの形成ができることなどから高密度記録の切り札的な存在ですが，歩留まりが少々悪く，値段が高いということもあって，コスト追求の激しいこの業界ではMIGヘッドを採用する場合が多いようです。

MIGヘッドというのは，通常のフェライトヘッドのギャップにあたるところに，フェライトよりも飽和磁束密度の高いセンダストをはめ込んだものです。薄膜ヘッドよりも安く作れるためローコストのディスクで使われていますが，いずれ薄膜ヘッドが量産体制に入るようになってくればその座を譲ることになるとみられています。

さて，一方の磁気ディスクの表層を成す媒体のほうはどうでしょう。小型のディスクドライブで長く使われてきたのは塗布媒体です。カセットテープなどでも同じような作り方をしていますが，要求される「薄さ」が一段と薄くなるにつれて均一に作るのが難しいことなどから，薄膜媒体と呼ばれるものに移行していくことになりました。

薄膜媒体というのは，金属の磁性体をメッキやスパッタリングなどの方法で円盤の表面にくっ付けてやるものです。これらの方法では，非常に薄く均一な磁性膜が形成できます。磁性体はメッキ媒体ではCo系の合金膜，スパッタリングではγ-Fe_2O_3やCo系合金膜が使われます（図5）。

薄膜媒体の開発で，磁性体の厚さは塗布媒体時の1/10以下，分解能は20%，出力は40%近く，SN比（信号，ノイズ比）は4デシベル以上の改善が行われました。

磁性体をくっ付けられる側である円盤の材料はアルミニウム合金が主流のようですが，アメリカのAreal Technologyの3.5インチハードディスクではガラス基板を使っています。トラック密度が上がってくると温度変化などによる寸法，形状の変化がヘッドのシークエラーのもとになりやすいので，ガラス基板に注目したのでしょう。

図4 薄膜ヘッドの構造

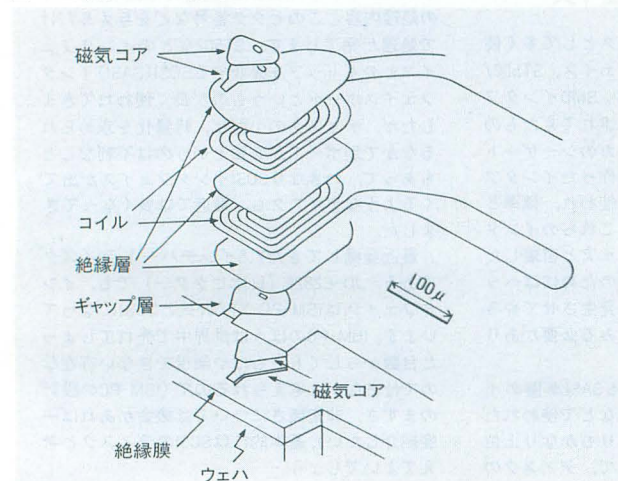
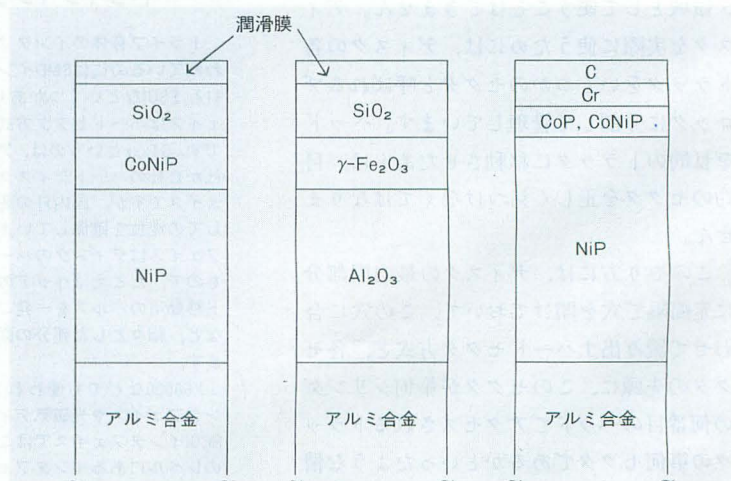



図5 薄膜媒体の断面例

第5章

SASIとSCSI

2つのインタフェイス仕様

SASI (Shugart Associates System Interface) は，アメリカのシュガート社が自社のハードディスク用に作ったインタフェイス仕様です。それをさらに進展させて複数のコンピュータでひとつの装置を共有する（マルチイニシエータ）ような機能や，同期伝送などの仕様追加を行い，ANSIで規格化を進めたのがSCSI (Small Computer System Interface) です。

通常，周辺機器というものは，フロッピーディスクならフロッピー専用のやり取りの方法があり，プリンタならプリンタのやり方があるために，それぞれ専用のインタフェイスが必要なのですが，SASI/SCSIでは，データのやり取りを行う方法を規定しているだけで，その先に具体的になにがつながっているかについてはまったく関知していません。使う側からすれば，バスの先になにがあるかは知らないが，とにかくブロック番号を指定して，そこから何ブロック読みたいよといえば，そこのデータが持ってこられることになります。つまり，SASI/SCSIインタフェイスでは，つながっているものはハードディスクでもフロッピーでも，ストリーマーやプリンタであっても，それこそX1やMZであってもかまわないのです。

SASIではひとつのバスに8台までのコントローラをつなぐことができるようになっており（SCSIでは自分自身も1台と数えるので7台になります），さらにそれぞれの先に8台までの機器接続を許すような仕様になっています。特定の機器にアクセスするときはまず，バス上の番号でコントローラを選んでから，そのコントローラに対して，そこの何番目のユニットにアクセスするかをコマンドと一緒に送りつけてやるようになっているのです。もちろん，この「ユニット番号」というのも，ホストから分かれて見えているだけの話ですから，相手側ではこっそりひとつのディスクのなかを2つに区分けしておいて，ホストには別々のユニットがあるように教えてしまうということもできます。

これをやっているのがカタログで「20Mバイト×2としても使えます」と書かれているものです。ホストはまんまとだまされて(？)，20Mバイトのディスクが2つあるのだなと思って処理してくれるわけです。

それではまず，現在X1，MZ-2500，X68000（PC-9801も）など，多くのパーソナルコンピュータで標準ハードディスクインタフェイスとなっている，SASIインタフェイスを見ていきましょう。ここで用意した相手側の機器は，NECのDS500B-100ディスクコントローラです。PC-9801用の純正ハードディスクであるPC-98H33（なんと10Mバイト！）に入っていたのは，確かDS501Bというコントローラでしたが，それのマニュアルを見つけ出すことはできませんでした。しかし，このDS500Bは型番こそ戻っていますが，時期的に考えてもDS501Bより新しいはずなのと（マニュアルの初版は昭和61年2月だが，そのころにはすでにPC-98H33は売っていた），いくらNECでもインタフェイスの仕様をコロコロ変えるほど愚かではないだろうと思い，これを引っ張り出すことにしました。

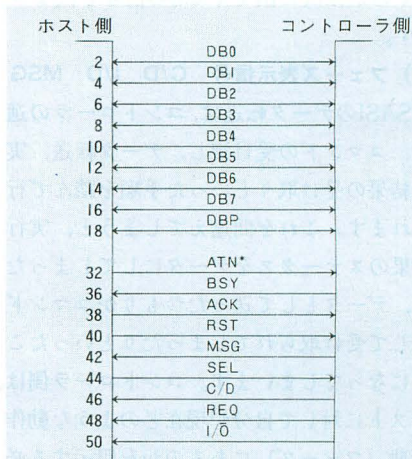
まず，SASIバスの信号を見てみましょう（図6）。意外と単純なので驚かれるかもしれません。双方向なのはデータバス（DB0-DB7）とパリティ（DBP）の計9本だけで，ほかの信号の方向は固定です。ホスト側から出力するのはACK，RST，SELの3本だけで，残るREQ，BSY，MSG，C/D，I/Oは入力信号になります。DS500BではSKEなる信号を勝手に追加していますが，これはまさに小さな親切大きなお世話の典型例といえるでしょう。

プリンタで標準となっている，セントロニクスプリンタよりもコネクタはひと回り大きい（セントロは36ピン，SASI/SCSIは50ピン）ですが，入出力信号の本数は大差ありません。これだけで，大容量デバイスと高速の入出力できるのです。

SASIの信号

1）データライン DB0-DB7 DBP

データやコマンドのやり取りのほか，コントローラを選択するために使います。DBPはパリティビットで，データが01なら0，F0なら1，といったように1のビットの数が奇数個になるように調整します。ピン切り換えなどでパリティを無視することもできるようになっています。SASI/SCSIバスでコマンドやデータの正当性を検証できる唯一の手段ですし，特にコマンドでは

図6 SCSI/SASIバス信号線

雑学への基礎知識

アクセス時間，平均アクセス時間

アクセス時間をさらに細かく分けてみると，目的のセクタのあるトラックまでのヘッドの移動時間（シーク時間），ヘッドが安定してデータの読み取りが可能になるまでの時間，そして目的のセクタがやってくるまでの時間（回転待ち時間）になります。当然，この時間は現在ヘッドのあるトラックやセクタの位置などに大きく影響されることになります。運がよければ，すぐ隣から読み出せるでしょうし，最悪の場合はディスクの最内周から最外周まで移動させられたうえにまるまる1回転分待たされることになるでしょう。

しかし，だからといって最良の場合と最悪の場合の時間をいわれても，あまり意味を成しません。まれには最悪値の保証が欲しいこともあるでしょうが，それよりも現実に使っていてどの程度速くなってくれるかのほうが重視されるべきでしょう。

次のアクセスで何ミリ秒かかるかは，具体的にいえなくても，ディスクにランダムにアクセスしたとき，平均してどの程度の時間になるかは先ほどの個々の時間がわかっていれば確率論を使って割り出すことができます。この時間を平均アクセス時間と呼んでいます。

一般的に，ハードディスクのカタログスペック上で，アクセス時間として公表されている数値は特に理由のない限り，この平均値のことを指しています。

1ビット間違うとまったく別のコマンドになってくれたりするのでなるべく使うようにしたいところですが，家庭やオフィスなどでは問題ないと考えているためか，パリティ回路分のIC（標準TTLでも3個くらい)をケチったためか，PC-9801ではパリティを無視してしまいました。このためハードディスク側でもパリティを無視するのが当然のようになって現在に至っています（諸悪の根源だなぁ98は)。

2）コントローラ選択信号　SEL　BSY

SELは，ホストがコントローラの選択動作を行ったことをコントローラに知らせるのに使います。このとき各コントローラはデータラインを監視して，自分が選択されたことを知るとBSYで応答します。

3）ハンドシェーク信号　REQ　ACK

REQは，コントローラがホストに対してデータやコマンドなどの転送要求をする信号で，ACKはホスト側からの応答信号です。ホストがデータを読み込むのか，データを書くのかは次に出てくるI/O信号で決まります。

4）フェーズ表示信号　C/D　I/O　MSG

SASIのデータ転送は，コントローラの選択，コマンドの受け渡し，データ転送，実行結果の受け取りといった手順を踏んで行われます。これを間違えてしまうと，実行結果のステータスをデータにしてしまったり，データとして送ったつもりがコマンドとして受け取られてしまったりといったことになってしまいます。コントローラ側は，ホストに対して自分が現在どのような動作状態（フェーズ）にあるのかを明示する必要があるわけです。このために用いられるのがC/D, I/O, MSGです。これら3つの組み合わせと，その内容は次のようになっています。各フェーズの意味などについてはあとで説明します。

C/D	I/O	MSG	
L	H	H	コマンドフェーズ
H	H	H	データ転送フェーズ
H	L	H	データ転送フェーズ
L	L	H	ステータスフェーズ
L	L	L	メッセージフェーズ

データ転送フェーズが2つあるのは，リードのときとライトのときがあるためで，リード時はI/OがL，ライト時はHになります。

見ればわかるように，C/Dはデータか否か（Hならデータ)，I/Oは転送方向（Lならコントローラからホスト，Hなら逆)，MSGはメッセージか否か（Lならメッセージ）を示すように割り振られています。

5）リセット信号　RST

SASIバス上の全コントローラにリセットをかける信号です。コントローラがディスクになにをしていようと（たとえディスクに書き込んでいる最中であっても)，一方的にリセットがかかることになるので注意深く使う必要があります。

SASIのフェーズ

SASIでは，バスの動作状態をいくつかのフェーズという単位に分けており，各フェーズを移行しながらデータ転送が進められていきます。フェーズの移行はSASIの動作を理解するための必須アイテムです。

それでは，SASIのフェーズの移行をディスクからのデータ読み出しを例にたどっていくことにしましょう（図7)。

1）バスフリーフェーズ（すべての信号線がH）

SASIバスがまったく使われていない状態です。誰もバスを使っていないときは，終端抵抗のおかげですべての信号はHレベルになります。出発点は必ずこの状態です。

2）セレクションフェーズ（SELがL，その後BSYがL）

ホストがSASIバス上のコントローラを選択するフェーズです。ホストはデータラインのいずれか1ビットをLにしてSELをLレベルにします。DB0がLだったら，ID番号が0のコントローラが，DB4がLならID番号が4のコントローラを選択することになります。各コントローラはスイッチ設定等で0から7までの異なるID番号と呼ばれる固有の番号を与えられていますので，各コントローラはその設定値と見比べ，選択されたのが自分だった場合にはBSYをLにして応答します。ホストはBSYがLになったのを見てセレクションの成功と見なし，SELをHレベルに戻します。BSYはコマンド実行がすべて終了するまでLになったままになります。

3）コマンドフェーズ（C/DがL，I/O・MSGはH）

セレクションフェーズで選択されたコントローラはC/DをL，I/O，MSGはHとして，ホストに対してコマンドフェーズになったことを示します。

コマンドの送出にはREQとACKの信号を使ったハンドシェークによって行います。SASIではコマンドやデータの区別なく，転送はすべてREQ－ACKハンドシェークで行われます。当然，転送方向によって2通りの動きが考えられるわけですが，コマンドフェーズではホストからコントローラの向きになります。このときのハンドシェーク手順は次のようになります。

・コントローラはREQをLにします。

↓

・ホストはコントローラに与えるコマンドをデータラインに乗せ，ACKをLにします。

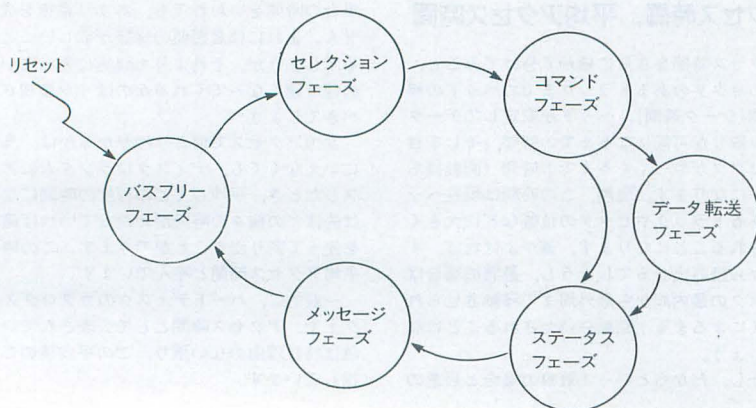
↓

・コントローラはACKがLになったのを見てコマンドを引き取り，REQをHに戻します。

↓

・ホストはREQがHになったのを見てACKをHに戻します。

↓

図7　SASIバスのフェーズ遷移

・コントローラはACKがHに戻ったのを見て1回の転送終了とします。

SASIのコマンドの多くは6バイトで構成されていますので，この手順を6回繰り返してコマンドの受け渡しが完了します。

4）データ転送フェーズ（I/OがL，C/D・MSGがH）

コマンド転送が終わったあと，コントローラはコマンドの解釈を行います。コマンドが，データの転送を必要としない場合には，すぐにステータスフェーズに移行します。データ転送を伴うものであった場合には，コントローラはC/DとMSGをH，I/Oはデータの向きがホストからコントローラ（ライト動作）ならH，逆にコントローラからホストならLになります。いまの例は，ディスクからのリード動作ですから，I/OはLになります。

データの受け渡しは1バイトごとに行われますが，これにはコマンドフェーズのときと同じようにREQ－ACKハンドシェークが使われます。データの転送方向が逆になる以外はほとんど同じになっています。

・コントローラはデータをデータラインに乗せてREQをLにします。

↓

・ホストはREQがLになったのを見てデータを読み，ACKをLにします。

↓

・コントローラはACKがLになったのを見てREQをHに戻します。

↓

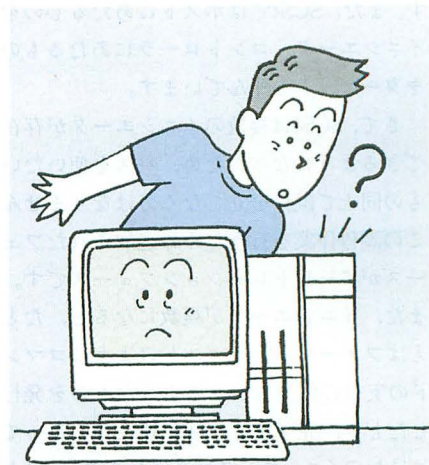
・ホストはREQがHに戻ったのを見てACKをHに戻します。

これはデータをセットする側が変わっただけでほかはほとんど同じです。

5）ステータスフェーズ（C/D・I/OがL，MSGがH）

データ転送が終了すると，コントローラはステータスフェーズに移行します。ステータスフェーズではコントローラは1バイトのステータスバイトをホストに送ります。この受け渡しもREQ－ACKハンドシェークで行われます。ステータスバイトではコマンドの実行が正常に終了したか否かを示すデータです。もし，エラーがあればさらに詳細なステータスを得るためのコマンドを送り直したり，エラーになったコマンドをリトライ（再試行）するなりの処理を行う必要があるでしょう。

6）メッセージフェーズ（C/D，I/O，MSGがすべてL）

一連の処理が完了したことを示すフェーズで，1バイトのメッセージバイトをコントローラからホストに転送します。DS500Bではメッセージバイトはいつでも0です。

SCSIのフェーズ

SCSIではハード的にはATN信号が増え，さらに平衡伝送用の仕様が作られるなどしました。フェーズのほうはアービトレーションフェーズ，リセレクションフェーズ，メッセージアウトフェーズが追加されました。さらに細かなタイミングなどを見ていくと，微妙に違うところがあります。特に，セレクションフェーズのタイミング，REQ－ACKハンドシェークのタイミングなどでは，最低限度確保しなくてはならない時間が規定されており，無制限に高速応答できたSASIとはずいぶん異なっていま

雑学への基礎知識

モーター

ディスクのヘッド移動の動力としてはステッピングモーター，直流モーター，ボイスコイルモーターの3種類にほぼ集約されます。

ステッピングモーターは，その名のとおりパルス電流を流すたびにカクカクとデジタル的に回るものです。与えたパルスに応じた回転角が得られ，割と簡単に精度が出ることからフロッピーディスクのヘッド移動用のモーターや小型のロボットの動力はほとんどこれが使われています。毎年，科学技術館で行われるマイクロマウス大会の参加マウスのほとんどは，ステッピングモーターを使っています。

直流モーターはブラシのあるものとないものに分けられます。ブラシというのは理科や技術科の時間に習ったような磁界のなかでコイルを回すタイプのモーターに電流を供給するために使うバネのことです。コイルの回転に応じて流す電流の向きが自動的に切り換わるため，回り続けられるわけです。構造は簡単，安価で，小型であるわりにはトルクが大きいのですが，なにせ回っている間，ずっとブラシがこすられているので自ずと寿命が限られてくるうえ，ブラシとコイルの接点で火花は出る，コイルに流す電流はすべてブラシとコイルの接点を通過するので，あまり大きな電流を流すとブラシとコイルの接点が溶けてしまう，低い回転数や一定速度での回転をさせるのは難しいなど，ブラシがあるが故の問題がたくさん出てきてしまいます。このような特性から，ブラシ付きのモーターは模型（いわゆる，マブチモーター）やカメラの動力など，長時間の連続回転を必要としないような用途に使われています。

ブラシレスモーターというのはその名のとおり，この諸悪の根源たるブラシを取り払ったモーターです。コイルのほうを固定して，コイルに流す電流をコントロールすれば，やはりモーターとして回転させることができます。AC（交流）モーターは基本的にブラシレスで，コンデンサを使って交流電流の位相を90度ずらしたものを使うことで磁場を回転させ，モーターはそれに合わせて回っています。身近なところでは扇風機のモーターがこのタイプです。

このAC電源の代わりに，直流電源をもとにトランジスタやICによって好きな周波数の電流を作り出し，モーターに与えるようにしたのがDCブラシレスモーターです。回転数を自由にコントロールできること，回転数を検出してフィードバックをかけ，回転数を安定化するなどの細工が割と容易であることなどが特徴です。機械的にも電気的にも静かで性質のよいモーターなので，CDプレーヤーを始め，長時間回転する必要のあるところにはほとんどのところでこれが使われています。

ボイスコイルモーターは，身近な例があまりありませんが，動作としてはスピーカー（ダイナミックスピーカー：最も一般的に使われているタイプのものです）を考えればよいでしょう。スピーカーの場合には，永久磁石のなかにボイスコイルと呼ばれるコイルが置いてあって，ここに流す電流に応じてコイルが動き，それに伴ってコーン紙が振動し，音を出します。

もしこれらを見たことがなければ，粗大ゴミとしてテレビやラジオの類が捨ててあったら，スピーカーをはずして分解してみるとすぐに納得できるでしょう。あとは強力な磁石として，スチール棚にでも張り付けておけば，メモクリップくらいには使えます（くれぐれもディスクと仲良くさせないようにご注意あれ）。

この「コイルが動く」ということをそのまま利用したのが，ボイスコイルモーターです。この構造をそっくりそのままひっくり返して，コイルのなかに永久磁石をおいて，磁石を動かすようにしたものはソレノイドと呼ばれ，こちらは鉄道模型のポイント切り換えや，大きなところではバルブの開閉制御などに使われたりしています。コイルを動かさないで済むという利点はありますが，なにせ磁石が重いので素早い電流変化に対する応答ではボイスコイルにかないません。

す。また，SCSIではホストにあたるものをイニシエータ，コントローラにあたるものをターゲットと呼んでいます。

さて，SCSIは複数のイニシエータが存在できるようになったため，バスを使いたいもの同士で調停を立てなくてはなりません。この調停作業を行うために追加されたフェーズがアービトレーションフェーズです。また，イニシエータが複数になると，たとえばフォーマットコマンドのようにコマンドの実行に何分もかかるようなものを発行したとき，完了までバスが使えなくなってはほかのイニシエータが迷惑します。このようなコマンドを発行したときは一度バスを解放して，バスフリーフェーズにしてほかのイニシエータにバスを明け渡し，実行が完了してからターゲット側が，自分にコマンドを渡したイニシエータを呼び出す方法がとれるようにしました。このフェーズがメッセージアウトフェーズです。

メッセージアウトフェーズでターゲットが自分を呼び出したイニシエータを認識するため，SCSIではイニシエータもID番号を持ち，セレクションフェーズのときに自分のIDも同時に出力するようにしています。

SCSIではこのほかにも，ACKを待たなくても，いくつのREQは一方的に送りつけられる同期モード伝送などがサポートされたりとだいぶ拡張が行われています。

このようにいくつかの違いはありますが，SASIはフルSCSIのサブセット版としてなんとか使える場合が多いようです。

ただ，細かく見ていくと，規格とはいいながら自由度が大きく，SCSI準拠とはいってもどこまで勝手なつなぎ替えができるのか不安を感じさせられる面がなくもありません。

コマンド概略

SASI/SCSIで使用されるコマンドは，まずグループ（またはクラス）という区分けがなされ，さらにそれが最大32種のオペレーションコードに分類されています。グループは0から7まででですから3ビット，コードのほうは5ビットで，両方合わせてちょうど1バイトとなっています。実際にコマンドとして送る場合にも，この両者を合わせて1バイトデータにして送っています。通常使用するフォーマットやリード/ライトといったコマンドはすべてグループ0になっています。SCSIでも同様です。

SCSIでは，グループ0以外のコマンドとしてはグループ1にコンペアとコピー&ベリファイがあるだけで，グループ2，3，4とグループ5のコード0からFがANSI標準の拡張用，グループ5のコード10から1Fまでと，グループ6，7は各社が自由に使ってよいコマンドエリアになっています。

コマンドのほとんどは6バイトで構成されています。フォーマット，リード/ライトなどもすべて6バイトです。SCSIでは6バイトで足りないときのために10バイトコマンドと12バイトコマンドのフォーマットを定めています。6バイトコマンドではブロック番号（セクタ番号）の指定が21ビットしかないので，光磁気ディスクのような大容量メディアでは完全に不足します。おそらく10バイトないし12バイトコマンドが用いられていると思われます。

DS500Bでは，クラス1に拡張リード/ライトと称して10バイトコマンドを追加していますが，ブロック番号は下位21ビットのみ有効としているので，ほとんど意味をなしていません。

これらのコマンドすべてについていちいち説明していては，Oh!Xの誌面を不法占拠した罪で「1年間の締切厳守」なんぞといぢめられそうなので，ここでは主要な5つのコマンドだけを紹介しておきましょう（グループはすべて0です）。これだけでも通常の使用では十分なはずです。エラーが発生したら，バスのRSTを使ってリセットして，リトライすればよいでしょう。

1）TEST DRIVE READY（コード00）

0	0	0	0	0	0	0	0
0	0	d	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

X：1/0のいずれでもよい
d：ドライブ番号

指定したドライブがレディ（使用可能）状態にあるかどうかをチェックします。電源

雑学への基礎知識

サーボ方式

ヘッドを正しく，目的のトラックに移動したかどうかをチェックし，ずれがあれば補正するのがサーボ回路の役目ですが，このチェックはディスク上に記録してあるサーボ情報をもとに行われています。サーボ情報を各トラックの先頭（インデックスサーボ）や各セクタの先頭（セクタサーボ）の情報に含めてしまうのをデータ面サーボ，ディスクの一面をまるごとサーボ情報専用面にしてしまうのをサーボ面サーボといいます。

今日のようにディスク枚数が1～2枚が普通になってくると，貴重なディスクの一面をまるまる占有されてしまうサーボ面サーボは，かなりもったいない使い方ということになります。ほかの方式にとって代わられることになるのが自然な流れだろうと思います。

欠陥セクタの処理

ハードディスクの全面がすべて健全で，完全な機能を果たせるというのは理想ではありますが，あれだけの精密機器ですので現実にはどうしてもデータの読み書きがうまくできないセクタが出てきます。このようなセクタを欠陥セクタ，欠陥セクタを含んだトラックを欠陥トラックと呼びます。フロッピーディスクでは容量もそれほど大きくなく，また取り外しが自由にできるので，欠陥セクタの処理を行わない場合も珍しくありませんが，ハードディスクでは欠陥セクタの処理は必修科目です。

欠陥セクタ/欠陥トラックの処理としては，あらかじめ予備のセクタ/トラックを設けておいて，欠陥があるとわかったときにそのセクタ/トラックを使用禁止にして予備の領域を使うようにする「代替えセクタ/トラック法」と，欠陥があるとわかった所をスキップする，「スキッピング法」の2通りがあります。また，代替えセクタ/トラック法では代替え領域を，ディスクの最終部分近辺に集中して配置する方法と，何トラックかおきに分散配置する方法が考えられます。

ほとんどのファイルが連続したセクタを使うようになっているとすると，アクセス時間が一番有利なのは「スキッピング法」，「分散代替えトラック法」，「集中代替えトラック法」の順になりますが，実際には「集中代替えトラック法」が用いられる場合が多いようです。これは，使っているうちに発生する欠陥（成長型ということもあります）に対処するのが容易であること，ほかの方法が割と面倒であることと，代替え領域を最終部分に集中させ，そこに一度移動してからまた戻ってくると考えれば確かに時間がもったいないようですが，確率とからめて考えるとほかの方法と比べてそれほど悪い値にはならないことが計算されるからでしょう。

代替えセクター法では，各トラックにひとつずつ予備セクターを設ける方法も考えられますが，あまりにも無駄が多いためにほとんど使われることはありません。

を投入したあとや，RST信号を使ってリセットしたあと，使えるようになったかどうかのチェックに使います。コマンドの2バイト目の00dとなっているところはユニット番号ですが，DS500Bではサポートできるのは2台までなので上位2ビットは常に0になり，dがドライブ番号を示すことになります。

このコマンドはデータ転送を伴わないので，コマンドフェーズのあとはすぐにステータスフェーズになります。ステータスフェーズで渡されるステータスバイトはコマンドの種類によらず，次のようなフォーマットになっています。

0	0	d	X	X	X	ERR	PE

上位3ビットの00dは，コマンドの2バイト目で渡したものと同じです。ERRはコマンドの実行が正常に行われなかったことを示すもので，1でエラーを示します。PEはパリティエラーで，DB0−DB7の8ビットのデータとパリティビット（DBP）のつじつまが合わないときに1にセットされます。ただし，PC-9801仕様ではパリティを無視しているので，これがセットされることはありません。

2）RECALIBRATE（コード01）

0	0	0	0	0	0	0	1
0	0	d	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
R	S	0	0	0	0	X	X

X：1/0のいずれでもよい
d：ドライブ番号
R：0のときリトライする。1ならリトライしない
S：0のときオーバーラップシーク禁止，1で許可です。1にしたあとはコマンド実行前にTEST DRIVE READYコマンドで，ディスクが使えるか（シークが完了したか）を確認しなくてはなりません

ヘッドをトラック0の位置に付けるとともに，ドライブ内部のエラーをクリアします。このコマンドに限らず，6バイト目のデータの指定がよくわからないときは0を入れておけば大丈夫です。

3）FORMAT DRIVE（コード04）

0	0	0	0	0	1	0	0
0	0	d	(MSB)論理アドレス				
論理アドレス							
論理アドレス(LSB)							
インタリーブファクタ							
R	0	P	F	BB		X	X

P：0のとき，フォーマットデータとして$6C_H$を使い，1のときには$E5_H$を使います
BB：上位ビットが0のときにはセクタバッファモードとなり，5バイト目のインタリーブファクタに従ってフォーマットします。上位が1，下位が0のときはトラックバッファモード，上位が1で下位が1のときはトラックキャッシュモードになり，いずれの場合もインタリーブファクタの指定は無効になります

指定した論理アドレスからドライブの最後のブロックまでフォーマットします。論理アドレスは，ディスク上の各セクタにディスクの先頭から順に割振った連続番号です。すべて0にすれば，ディスクの全面をフォーマットすることになります。

4）READ（コード08）

0	0	0	0	1	0	0	0
0	0	d	(MSB)論理アドレス				
論理アドレス							
論理アドレス(LSB)							
ブロック数							
R	E	0	0	BB		X	X

E：0のときECCによるデータ修正を許可します

指定した論理アドレス番号のセクタから，ブロック数で指定したセクタ数分だけデータを読み出します。Eビットはディスクへの書き込み時に付けておいたECCコードによって，データにエラーが発生したときに，自動的にデータの修正を行うことを指示するもので，通常は0にしておいたほうがよいでしょう。

このコマンドには，データ転送フェーズが伴っています。

5）WRITE（コード0A）

0	0	0	0	1	0	1	0
0	0	d	(MSB)論理アドレス				
論理アドレス							
論理アドレス(LSB)							
ブロック数							
R	0	0	0	BB		X	X

指定した論理アドレス番号のセクタから，ブロック数で指定したセクタ数分だけデータを書き込みます。

このコマンドもデータ転送フェーズを伴います。

やっぱり奥は深い

あまりにも高価でパーソナルとは口が裂けてもいえなかったために，取り上げる機会のなかったハードディスクをついに攻めることになりました。当初はあまり突っ込む必要はないだろうとタカをくくっていたのですが，本気で調べていくとそんなに甘いものではありませんでした。ハードディスクというのは電磁気学，機械工学，制御工学，電子工学，情報工学など，物作りに絡む技術，それもかなり高度な部分の集合体だったのです。本気でやっていったらハードだけで最低2巻，SASI/SCSIではどう考えてもまる1巻，さらにアプリケーションなどでの応用だけでも1巻となって本屋に並んでいることを考えると，「雑学ハードディスク全何巻」なんていうことになってもおかしくないくらいです。

今回はなるべく広く，しかもところによっては少し深くと，アサガオの種まき式にいろいろなところをつついてみましたが，いかがだったでしょうか。なにか興味が出てきたら教科書なりそれなりの専門書を当たってみると良いでしょう。理工学系の大学があれば，そこの生協などで買えるでしょう（生協の会員でないと割引はしてもらえないだけのことで，誰でも買うことはできるはずです）。

とにかく，このながーい解説に最後までお付き合いいただいて，お疲れ様でした。

HARD DISK

超初心者に贈る

HOW TO USE HD

Nishikawa Zenji
西川 善司

ハードディスクはまったく初めてという西川氏がついに1台のハードディスクを手に入れました。マニュアル片手にあーでもないこーでもないと大奮闘。初心者の目から見たハードディスク入門です。

初めてのハードディスク

これまで隠してきたけれど実は私こと西川善司は初体験なんです。なんの初体験かってそれはタイトルを読めばわかりますよね。そう，私は今まで松茸を食べたことがなかったのです，じゃなかったハードディスク(以下HD)を持っていなかったんです。ところがこの間秋葉原へフラフラッと立ち寄ったらHDの安いこと安いこと。昔は10MバイトのHDは20～30万しましたが，今では40Mバイトが10万円を割っているという……。私は驚きのあまり，駅前の客引きのおじさんの後ろについて行ってしまい，ふと我に帰ったときには40MバイトのHDを買わされていました。うぅ，明日から当分ご飯はペヤング(注1)だな，と思いつつ帰路についたのでした。

HDをフォーマット

うししっ，セッティングはこれでOKだ。昨日までBASICプログラムをコンパイルするのに，いちいちCコンパイラのディスクのマスターをコピーして，いらないファイルを消してディスクの容量を稼いで，ふぅ，それからやっとソースをコピーしてコンパイルしてましたからね(X68000のCコンパイラはコンパイルの途中，いろんなファイルをディスクに作るから，コンパイルするソースの大きさに応じたある程度の空きエリアが必要なのだ)。今日からそんなことをしなくて済みそうだな，などと思いつつHuman68k ver.1.00を立ち上げます。HDも普通のフロッピー同様フォーマットしなければ使えないみたいです。えーと，マニュアルのFORMATの項を見てみるとハードディスクのフォーマットは，

FORMAT /H

とすればいいみたいですね。

?：ふふふ，そのとおりじゃ。

善：だ，誰だ!?

老：はっはっは。ここじゃ。(ギィーっ)

善：この暑いのによくタンスに隠れてましたね……。しかし，どうしてここに。

老：いや，暇だから西川君のところにきてみたのじゃ。しかし，不用心じゃのう。留守なのに鍵ひとつ掛けとらん。

善：忍び込んどいてなにいってんです。

老：そんなことより，X68000を見てみなさい。ほら，なにか聞いてきとるぞ。

ハードディスクの装置番号を指定してください。[0～15]：

善：ふーむ，するとX68000は本体にハードディスクを16台も接続が可能なのか。うーん，すごい。えー，1台しかないから0と押して，と。

ハードディスクのタイプを指定してください。(10，20，40Mバイト)＝

老：今日おぬしが買ってきたのは40Mバイトのハードディスクじゃから，40と入力すればいいのじゃ。

善：んなこと，見りゃわかります。

老：ほら，またなにか聞いてきとるぞ。

装置全体を初期化します，よろしいですか？〈Y/N〉

善：Y。おお，HD正面のランプが赤くなった。フロッピーと違ってかなり待たされますね。

老：それもそうじゃ。フロッピー40枚分も容量があるんじゃからな。

初期化を終了しました。

善：やっと終わった。これでHDが使えるようになったわけか。

老：いやいや，まだじゃ。

処理を選択してください。
1：マップ　2：領域確保　3：領域開放
4：領域選択　5：装置初期化　6：装置変更　7：終了

善：先ほど行ったのはメニューの6と5に当たるわけか。で，これからあと，なにをすればいいというんですか？

老：2の領域確保を選びなさい。

何メガバイト確保しますか？：

善：40と。

老：待ったぁーっ!!　確かにそれでもいいが，HDは1台であたかも2台以上のものとして使うことができるのじゃ。

善：へぇ。

老：いま，おぬしのHDは40Mバイトじゃから，ここで，20Mバイトの領域確保を2回行えば，1台のHDで20Mバイト2台分のHDとして使えるのじゃ。

善：ふぅん。それじゃ10Mバイトで4回領域確保すれば1台で，4台分として使えると？

老：そういうことになるな。で，そのようにして分けられた1つひとつの領域を「ブロック」と呼ぶ。

善：それじゃ，20と入力しておくか。

領域確保中です……

領域確保ってのはずいぶん時間のかかるものなんですね。

善：うんと，複数のブロックに分けたとして，それはどんなふうに使うもんなんですかね。

老：そうじゃな。X68000にはHuman68kのほかにもOS-9というOSがあるから，ブロック1はHuman専用，ブロック2はOS-9専用の領域として使ったりするのが一般的かのう。

善：あ，なるほど。ブロック1はHuman68k ver.1.00で，ブロック2にはver.2.00を入れておく，なんてこともできるんですかね。

老：もちろん，できる。ほら，領域確保が終了したようじゃぞ。

IPLの登録およびFAT・ディレクトリの初期化を行います。

ボリュームラベルを半角21文字以内で指定してください。

善：ええと，んじゃ，Human68kと。

システムを転送しますか？〈Y/N〉

善：Y。ほんじゃ，ま，メニューの7の終了を選んでと。

ハードディスクの初期化を終了します。

一度リセットしないと，ハードディスクは使用できません。

リセットしてもよろしいですか〈Y/N〉

善：こ，これでやっと私のHDは，使えるようになったんですね。よし，リセット!!

うう，なんて速くHumanが立ち上がるんだ。うううう……。

HDにディスクをコピー

善：うううう。

老：いつまで泣いておるんじゃ。

善：いや，明日から毎日食事がペヤングになるかと思うと妙に泣けてきましてね。

老：そうか。(アッサリ) さて，いまHDにはHUMAN.SYSとCOMMAND.Xしかないわけじゃが，これだけではなにもできない，違うか？

善：そうですね。VS.X(ビジアルシェル)なんかもほしいしFM音源ドライバなどのデバイスドライバなんかもほしいですよね。あ，わかった！ このシステムディスクをHDにコピーすればいいんですね。

老：それはそうだがどうやってコピーしようというんだね。

善：まぁ，見ててください。リセットして，このシステムディスクのマスターを立ち上げます。……VS.Xが立ち上がった。いまこのシステムディスクが入っているのはドライブA，HDはドライブCになってるから普通にディスクをコピーするみたいにディスクのアイコンをA→Cへコピーすればいいんでしょ。

老：やってみんしゃい。

善：ふん，偉そうに。えい，あれ？

メディア/容量が違います

コピーできません

老：はっはっはっ。それみたことか。

善：笑ってないでどうしたらいいか教えてくださいよ。

老：このディスクを立ち上げてみなさい。

善：なんじゃ？ こりゃ。ただのHuman68k ver.2.00じゃないですか。

老：はっはっはっ。あいかわらず勉強不足じゃな。ver.2.00にはCOPYALLという便利なコマンドがあるのを知らんようじゃな。こいつは種類の違うメディア間のオールコピーができるコマンドじゃ。

善：どうやって使うんです？

老：いま，ドライブCはHD，Aドライブには先ほど立ち上げたver.2.00が入っておる。ドライブBにコピー元となるおぬしのシステムディスクを入れなさい。

善：はい。

老：ここで，

COPYALL /T B：*.*C：

を実行してあとは待つだけじゃ。

善：「B：*.* C：」はだいたいわかるんですけど「/T」ってなんです？

老：COPYALLコマンドのスイッチでコピー先にすでにあるファイルはコピーしない，ということじゃ。ほら，HumanやCOMMAND.XはすでにHDにFORMATのとき入れたからコピーする必要はないじゃろ。COPYALL にはほかにもスイッチ指定ができるからマニュアルで確かめておくといいじゃろう。

善：ふつうのCOPYコマンドとどう違うんです？

老：まぁ見ていなさい。ほら，COPYALLはサブディレクトリの内容までコピーしてくれる。

善：おぉ。コピー先にないディレクトリは自動的にMKDIRコマンドを使って勝手に作ってくれてるぞ。なんて賢いやつ！

老：このほかにもver.2.00は便利なコマンドが増えているから購入するなり(税抜き9,800円)，EXPERT，PROに買い換えるなりしたほうがよいぞ。試しにFORMATコマンドを起動してみなさい。

善：おー，さっきまでの操作がメニューで出てきた。これは楽チン。どうして早く教えてくれなかったんです？

ver.1.00とver.2.00との違い

善：もう一度20Mバイトの領域確保をしたらブロック1とブロック2に分かれるんでしょ？

老：そうじゃ。

善：X68000はブロック1，ブロック2，どちらを起動していいか迷いませんかね。

老：なにをたわけたことといっておる。ちゃんとIPLがどれを立ち上げるかを聞いてくるようにできておる。

善：あ，本当だ。

老：これはver.2.00でフォーマットすると出てこないメニューじゃ。もともと，OS単位で起動を選択するためのものじゃからな。

善：さっきver.1.00とver.2.00で分けて使えるといってましたが。

老：やればできる。まあ，あまりメリットがあるとは思えんが……。まず，ver.1.0でフォーマットして領域を2つ確保する。ここではシステムを転送せずに，それぞれのシステムディスクを立ち上げ，SYSコマンドでシステムを転送すればいいんじゃ。

善：これで起動時にOSを選択できるわけですね。で，選択されなかったほうのブロックはどうなるんです？ もう使えなくなっちゃうんですか？ 領域選択されなかったほうのブロックにはアクセスできないんですか？

老：ほう。おぬしにしてはまともな質問じゃな。結論からいうとHuman68k ver.1.00ではアクセスできん。電源ON時のハードディスクメニューで選んだブロック，もしくはFORMAT /H の領域選択で選んだブロックにしかHDとしてアクセスできない

図1

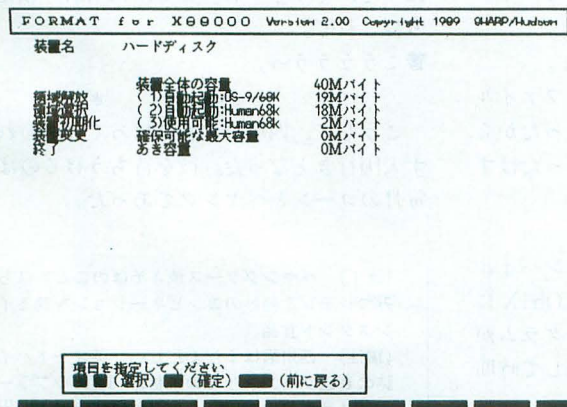
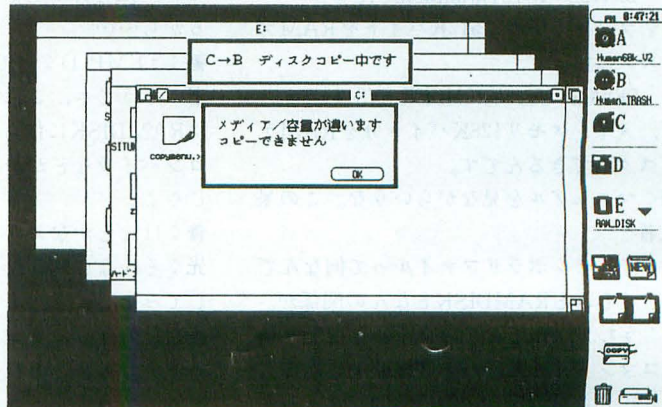
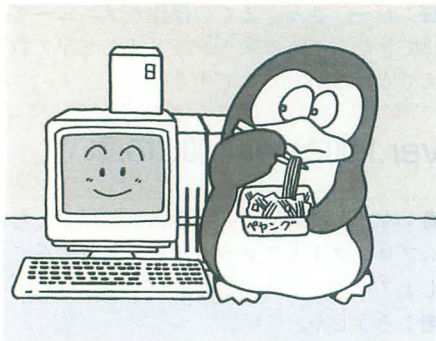



図2

のじゃ。
善：その口振りからするとver. 2.00からはできるんですね。
老：そうじゃ。だから，HDのブロック間でのファイルのやりとりも可能なのじゃ。

テンポラリファイルの秘密

善：いやぁ。HDはいいなぁ。BASICがものの2秒で立ち上がる。うう。OPMAドライバの組み込みもまったく待たされないし。
老：そうじゃ，おぬし，テンポラリファイルを知っておるか？
善：は？ あの金属製の四角のワクに布をはって，跳躍運動に使う運動器具ですか？ うーん，そういえば昔ドリフの「全員集合」とかで使ってましたね。あれがどうしたんですか？
老：長々と説明ご苦労。それはトランポリンじゃ。わしがいってるのはトランポリンファイル，じゃなかった。
善：テンポラリファイルでしょ。
老：……。
善：で，それがどうしたんです？
老：おぬしはいつもそんな調子か？ まぁよい。で，おぬしは自分のシステムにRAMDISKを作っとるかな？
善：いや。でも作り方は知ってますよ。CONFIG. SYSで，

DEVICE=¥SYS¥RAMDISK.SYS #G

でグラフィックラム512KバイトをRAMディスクにできて，

DEVICE=¥SYS¥RAMDISK.SYS #M128

で，メインメモリ128Kバイト分をRAMディスクにできるんです。
老：マニュアルを見ながらいうな。この呆気者！
善：で，テンポラリファイルって何なんです？ それとRAMDISKとなんの関係が。
老：うむ。SORTなどのフィルタ機能を持つコマンド，まあ，パイプ機能というんじゃが，これらの実行の際，中間ファイルを作るんじゃ。
善：それをテンポラリファイルというと？
老：そうじゃ。コマンド終了後このテンポラリファイルは自動的に消される。アセンブラやコンパイラなども動作中テンポラリファイルを作ったりするのじゃが，これらのプログラム（コマンド）の処理時間の多くはこのテンポラリファイルの作成に費やされておる。
？：ふふふ，そのテンポラリファイルを速いメディアに割り当てることができればコマンドの動作時間を短くできるというわけです。
善：だ，誰だ!?
光：はっはっはっ。ここです。トイレを借りていました。西川君，ちゃんとトイレは使用後流すようにね。あれでは女性に嫌われますよ。
善：よけいなお世話だ。どいつもこいつも。
老：いや，悪い悪い。つい講釈に熱が入りすぎて彼も一緒だというのを忘れとった。
光：さて，西川君。HumanにはTEMPという内部コマンドがありましてね。テンポラリファイルをどこに作るかを指定できるんですよ。
老：いま立ち上がっとるシステムはHDから立ち上げたものじゃからテンポラリファイルはHDに暗黙のうちに作られとる。まぁ，これでも十分速いのだが……。
光：RAMDISKの速さには及ばないというわけです。便利だからといって，なんでもかんでもハードディスクにやらせるのも考えものでしょう。
善：なんだい。2人してしゃべりまくって。
老：まあ。見ていなさい。CONFIG. SYSに，

DEVICE=¥SYS¥RAMDISK.SYS #G

を付け加えてリセットし，システムを立ち上げなおす。
光：まず，HDがAドライブ，フロッピーディスクドライブ（以下FD）がB，Cドライブ，RAMDISKがDドライブとなっているから……。
善：TEMP D：っていうんでしょ。
老：そうじゃ，これでテンポラリファイルはRAMDISKに作られるようになったから，コンパイラなどがかなり高速になったはずじゃよ。
善：ほんとかなぁ。
光：そんなに疑うなら，なにかコンパイルしてみましょう。そうだ，今月のOh!Xにあなたの作ったミュージックプログラムがありましたね。あれをコンパイルして時間を計ってみましょう。

CC ファイルネーム.BAS

と。
老：以下にコンパイル時間を示しておくから読者はテンポラリファイルはFDよりはHD，HDよりはRAMDISKに作らせたほうが速いということを確認してほしい。
善：老師，誰に向かって話してんです？

テンポラリファイルを，
FDに（コンパイル時間123秒）
HDに（コンパイル時間93秒）
RDに（コンパイル時間69秒）

HDよ，永遠に

光：まあ，西川君もHDユーザーとなったわけだから，お祝いにZ80's Barにでも飲みにいきますか!?
善：そんじゃ光のおごりということで。ルンルン……おっ，地震だ……。

――ガッシャーン――

うわぁーっHDが落ちた。どうしたらいいんじゃー。うううっ。
老：落ちるのも当たり前じゃ。なにせディスプレイの上に置いてあったのじゃからな。HDはな，振動やショックには非常に弱いのじゃ。
光：書き込み中に揺らしたりしても，クラッシュするくらいですからね。
老：HDは運搬にも気をつけなければならん。HDは基本的に，ヘッドとディスクは非接触でアクセスしておるから，振動によってヘッドが強くディスクにぶつかったりして，ディスクに傷をつければ故障となりかねない。
善：2人ともそんなに知っているんだったらなんでひと言，いってくれなかったんだよー。うわー。102,794円(税込み)が一瞬にしてパーだ，あー，明日からペヤングだというのに……。
老：あー，飲みには2人で行くとするか，光君。
光：そ，そうですね。じゃ，西川君。私たちはこれで。
善：うううううっ。

＊　＊　＊

こうして，西川君のHDはろくに使われず天国行きとなった。彼を待ちうけるのは毎月のローンとペヤングであった。

(注1) ペヤングソース焼きそばのこと。はちみつレモンと絶妙のコンビネーションを誇るインスタント食品。
(編注) 西川君はまだまだ甘い。某オートバイ誌の貧乏グランプリでの名言，「3食カップラーメンは貧乏じゃない。カップラーメンは128円もするじゃないか！」

HARD DISK

無難に分割統治を行うための

空間有効利用の心得

Ogikubo Kei
荻窪　圭

使い慣れた2Mの世界から，ハードディスクを手にすることによって広がる大容量の世界。これは，いきなり膨大なメモリ空間を前にしても，臆することなく勇猛果敢にチャレンジするための，心温まるユーザーズマニュアルなのです。

Human68k ver2.0の紹介記事（Oh!X 5月号）を書いていたら，無性にハードディスク（注1）がほしくなり，とうとう，ゴールデンウィークに突入するや否や秋葉原に飛んでしまった。右手にクレジットカード，唇に「ボーナス一括払い！」。

ここで重要なのは，“ゴールデンウィークを待った”ことである。誰にも（締め切りにも）邪魔されず，ゆっくり40Mバイトを堪能する時間を必要としたからだ。40Mを自分の好きなように使う。ほとんど気分は都市計画。ああ，40Mは広いなあ，とりあえず。うちのハードディスクは未だ10Mほどの原野を抱えているんだもん。

第1部 生のHDを前にして石になる

まずは使えるようにする

その昔，ハードディスクというのはお金を持ったパソコンユーザーや，貧乏でもパソコンには金を惜しまない剛の者や，お仕事にパソコンを使う“買ったハードディスクの代金が必要経費として減価償却できる”人のものだった。つまり，自分で“都市計画”を遂行できる人や，ロータス1-2-3なんかのビジネスソフトしか使わない人だ。

ところが，最近では困ったことに，“パソコンを買ったらなかにハードディスクが入っていた”などというラッキーな人々が増えている。ラッキーなだけならまだいいが，生まれて初めて買ったパソコンがX68000 ACE-HDで，ハードディスクがなんたるかさえ知らなかった，となるとこれは原野商法にひっかかった老人のようなものだ。

だから，今回は原野の開拓，戦う屯田兵，土地は狭くとも磁性面は錦の精神で追ってみようと思う。最初の教えは“広がる磁性面で石になる者は幸いである”だ。

初めてX68000でHDを使う

まず，パソコン初心者に贈るハードディスク“使い方”入門から始めよう。買ってきて本体をじっと見つめ，“ハードディスクとやらはいったいどこに入っているんだろう”と悩んでしまうような人である。つまり，初めっからハードディスクが内蔵されているACE-HDやらExpertやPROのHD版を買った人ね。もしかしたら，インストールしないとハードディスクはなんの仕事もしてくれないことも知らないかもしれない。

で，仕方がないから，フロッピーディスクとハードディスクの違いもわからぬまま，マニュアルを見てエイヤっと“ハードディスクインストール用”バッチファイルを実行してしまったりするかもしれない。それでもって，買ってきたゲームで遊ぼうとドライブ0にディスケットを入れたはいいが，ハードディスクからいつものようにビジュアルシェルが立ち上がったりして“どーして僕のハードディスクはいつもしゃしゃり出てしまうのだ。ゲームができないぞ！”と叫ぶかもしれない。

それは，ハードディスクとX68000(Human)の関係を正しく把握していないからである。マニュアルにいわれるままハードディスクをフォーマットし，システムディスクをコピーしてはいないか。ハードディスクの20Mなり40M（注2）なりの広大な空間（磁性面）はユーザーが管理するあなただけの土地（しかも整地前の原野）であり，ただ単に“広いからファイルがたくさん入っていーなあ”というものではないのである。40Mもどうやって使おうかなあ，と考えるだけで2，3日暇が潰せてしまうくらいのものなのである。ハードディスクのインストールは愛を込めて，100坪の土地があったらどんな家を建ててどんな部屋割りにしてどこを誰の部屋にして（私は2階の8畳くらいある洋間で，ベランダから天気のいいときには富士山が見えるといい）などと想像するような心で行うべきなのだ。

ハードディスクの構造やら選び方やら壊し方やらお風呂の入れ方などそういったことはほかの記事に任せておいて，私はハードディスクにおける“都市計画”，もう少しスケールを落とすなら，FDからHDへの楽しいリハウスを追求してみたい。

それにはまず，OSを用意する。これは重要である。X68000にはいま，4種類のOSがある。Human68k ver1.0とver2.0とOS-9/68000とCP/M68Kである。ここはとりあえず初心者には関係のない（はずの）OS-9/68000とCP/M68Kは省いておこう。すると，残りはver.1と2の2つだけだ。ここで異論もあるだろう。ver.1と2を分けなければならないのか。そうである。FDしか使わないのなら困らないが，ことハードディスクに限ってはこれはもう，かわいそうなくらいHuman68k ver1.0は過去のものなのである。悪いことはいわない。HDをHuman68k ver1.0で使うのは特別な理由がない限り遠慮したほうがいい。ver.1で使っている人はver.2（もちろん買ってくるのだ。下手なゲームよりずっと高いコストパフォ

注1） HDだとハードディスクだけど，2HDだと2ハイデンシティになる。うーん，略語って微妙。

注2） 広大な空間だが，広さに紛れて“不良セクタ”や“不良クラスタ”などがよく潜んでいる。大容量の精密機械の場合などには多少の落ちこぼれが存在するものだ。ただ，そういった事態を想定して，大きめに容量が用意されたりするので，さほど心配する必要はない。

ーマンスで楽しめるぞ）で，インストールし直したほうがいい。ver. 1と2では建売住宅と注文建築くらいの違いがあるのだ。

理由は簡単である。Human68k（に限らずたいていのDOSはそうだ）はひとつのハードディスクをいくつかに分割して使える。40Mを20Mのハードディスク2つと見なして使うとかね。このとき，ver. 1ではそれぞれ同時に使うことはできないが（立ち上げ時にどちらを使うか決まってしまう），ver. 2では領域（ブロック）を確保しただけ別のドライブとして使えるのだ（図1参照ね。これがいまの私のハードディスクの使い方）。どうだ。わがままだろう。たとえば，40Mのドライブがひとつあるのと，10Mのドライブが4つあるのとではどちらが扱いやすいか？

これが一番大きな理由である（そういえば，この話は5月号の“Human68k ver.2詳解”ではし忘れたな。当時，ハードディスク持ってなかったしな，と言い訳をしておこう）。そんなわけで，持っていない人にはHuman68k ver.2を買うよう勧めるのである。その他もろもろハードディスクに親切な設計だしね。

で，話は戻って，ハードディスクのインストールであるが，まずはフロッピーディスクと同様，フォーマットしてやらねばならない。ver. 2の場合はFORMATコマンドを実行するだけでいい。あとはメニューが出るから，そこでハードディスクを選択して，思い切ってエイヤッとばかりに初期化を選べばいい。装置番号は0でね（ハードディスク内蔵可能機種にもかかわらず，外付けを買った人を除いてだけど）。ver. 1の場合は，コマンドモードで，“FORMAT/H”と入力する。

ここでエイヤッと実行して，“システム管理領域が異常です”といわれても，まっさらなハードディスクは異常なのが当たり前なわけだから，びびってはいけない。軽い気持ちで，そのハードディスクの容量を入れてやる。

ハードディスクのフォーマット作業（図2，3のような状況で行う）は，いかにフロッピーディスクよりアクセスが速いとはいえ，何10倍も容量があるわけだから，時間がかかる。本当に時間がかかる。じっと見ていると不安になってくるので，ディスプレイをテレビに切り替えるなり寝るなりOh!Xの愛読者カードを書くなりするのがいい。エッチだってできるぞ。そして，待つ，待つ，待つ。

やがて，終わる。ハードディスクはフォーマットしただけでは使えないのだ。

フォーマットが済んだら，領域確保という“広大なハードディスクのうち，これこれこれだけをひとつのドライブとして使いたいから，区画整理をして管理領域を作ってちょ！”宣言が必要なのだ。そして初めてハードディスクがドライブの一形態だと認めてもらえるのである。

その領域の確保だが，きちんと“都市計画”ができていないとユーザーは悩むことになる。特にver. 2では，あなたの一存で40Mなり20Mなりがいくつものドライブになる。つまり仮想ドライブ（注3）ではなく，れっきとした物理的なドライブになるのだ。X68000を前にして石になるところといえよう。私は軽い気持ちで20Mと10Mの2つを確保して残りを原野のまま放っておいたのだが，結局あとから領域確保をやり直す羽目になってしまった。きめ細かな領域確保が肝心なのだ。

さて，どれだけ確保しておこうか。ver. 1なら，でっかくとればいいのだけれど（どうせ一度に1カ所しか使えないのなら，全部とっちゃえ！），ver. 2となると1Mずつ40個とか（いくらなんでもそれは無茶だ。ドライブ名が足りない。A～Zまでだから，

注3） ディスクドライブそのものとは違って，ひとつのディレクトリなどを論理的にディスクドライブとして扱うこと。たとえば「“A:¥BIN¥”をGドライブにするよ」と宣言すると，“DIR G”で“A¥:BIN¥”のディレクトリを見ることができる。

図1 DRIVE /V

```
A：2HD（1MB）      ユニット番号……0
B：2HD（1MB）      ユニット番号……1
C：ハード ディスク ユニット番号……0ボリュームラベルはHuman2_SYSです
D：ハード ディスク ユニット番号……1ボリュームラベルはツール＆ゲームです
E：ハード ディスク ユニット番号……2ボリュームラベルはデータです
F：ハード ディスク ユニット番号……3ボリュームラベルは辞書です
G：RAM DISK        ユニット番号……0ボリュームラベルはRAM_DISKです
H：仮想ドライブです
        ⟆
Z：仮想ドライブです
```

図2 Human ver1.0

```
Format version 1.02
ハードディスクの装置番号を指定してください[0～15] :0

★★★★  システム領域の管理情報は次の通りです  ★★★★
装置全体の容量            40メガバイト
ブロック( 1):Human68k     40メガバイト
あき領域    :              0メガバイト

処理を選択してください
1:マップ 2:領域確保 3:領域開放 4:領域選択 5:装置初期化 6:装置変更 7:終了 =1

★★★★  システム領域の管理情報は次の通りです  ★★★★
装置全体の容量            40メガバイト
ブロック( 1):Human68k     40メガバイト
あき領域    :              0メガバイト

処理を選択してください
1:マップ 2:領域確保 3:領域開放 4:領域選択 5:装置初期化 6:装置変更 7:終了 =■
```

図3 Human ver2.0

```
FORMAT for X68000 Version 2.00 Copyright 1989 SHARP/Hudson

装置名        ハードディスク
装置番号      0

領域確保        装置全体の容量              40Mバイト
領域解放        ( 1)自動起動:Human68k       40Mバイト
領域選択        確保可能な最大容量           0Mバイト
装置初期化      あき容量                     0Mバイト
装置変更

項目を指定してください
（選択）（確定）（前に戻る）
```

まあ，多くて2Mずつの20個だな)，5Mと10Mと15Mの組み合わせでスウェーデンリレーなんてのもできる。やっぱ，ver.2だよ。うん。

とりあえず領域確保をすると（これもまた時間がかかるから，コーヒーくらいは飲める。ジャワティなら1リットル瓶が飲める，かもしれない)，あとは，システムを転送して，リセットすれば大丈夫。この，リセットというやつが重要で，OSさんは本体を立ち上げ時につながっていた（確保されている）ものでないとドライブとして認めてくれないのである。

細かい話，たとえばどう領域を選ぶといいかとか，ハードディスクになにを入れればいいの？ なんかはあとで考えることにして，次である。

外付けHDの場合

外付けだってまあ，内蔵と違いはないんだけれど，電源ケーブルとハードディスク用のケーブル（これがまた太いんだ）があるだけでね。こいつをつないで，前項と同じことをすればいい。ただ，ハードディスクの電源を入れてから本体を立ち上げなければならないとか，本体の電源を落とす前にハードディスクを切ってはいけないとか（先日，なにを狂ったかハードディスクのパワースイッチを押してしまい，この指を離してはいけない！ と，右手人指し指でハードディスクのスイッチを押したまま左手で本体の電源を切った）あるけどね。

ただ外付けの場合，不必要なとき（ゲームをするときとかね）には電源を切っておけるから安心だ。ハードディスクというのは電源の入っている間中ずっと回りっぱなしだから，科学技術を信頼していない私としては，使わないときは止まっていてほしい。ついでに，ハードディスクがぶっ壊れて（アクセスしているときに大地震が来てしまって，揺れた拍子に磁性面を傷つけてしまったとかね）使えなくなったとき，外付けなら外して修理に出せばX68000はそのまま使えるけど，内蔵タイプだとそのまま本体ごといなくなってしまうから，そういった点でも有利だ。内蔵か外付けかは趣味の問題(ウソ，機種の問題だ)。私のX68000は初代機種だから，内蔵はできないのさ。

で，どんなのを選ぶかだけど（シャープは純正の外付けの40Mハードディスクを出していないからサードパーティものに頼ることになる)，うちのヤツみたいに安いけれどBREAKキーでシッピングしてくれないのや（でも，電源OFF時の自動シッピングはあるから安心)，2台目のハードディスクをつなぐ出力端子があるのやないのがあるから，今月の特集をよく読んでから考えよう。大きなグラフィックデータをいっぱい使う人は2台目がつながるヤツを買いたいだろうし，そういった予定がない人は少しでも安いほうがいいだろう。

で，持っていない人は買った自分を想定して次へどうぞ。

フォーマットのお次の作業

フォーマットと領域確保とシステム転送がすんだら，ハードディスクから立ち上がるようにしなければならない。FORMATコマンドが転送してくれる「HUMAN.SYS」と「COMMAND.X」のほかに，「うんちゃら.SYS」シリーズも必要だろう。簡単にすますには，ACE以降の機種に付いているハードディスクインストール用のバッチファイルでいいかもしれないが，これは次の点でよろしくないのである。

1）システムディスクをそのままコピーしてしまうので，環境作りを味わえない。あるいは環境作りの勉強にならない。
2）勝手にメモリスイッチを書き換えてしまう。
3）勝手なCONFIG.SYSとAUTOEXEC.BATを作られてしまう。

なかでも特に2)が恐怖だ。なんと，なにがなんでもハードディスクから立ち上がるように書き換えてしまうのである。どういうことかというと，フロッピーディスクドライブにデータディスクがささっていようがゲームがささっていようが，ハードディスクからブートしてしまうのだ。フロッピーディスクで立ち上げようと思ったら，OPT.1キーを押しながら立ち上げなければならない（注4）。困ったもんだ。

ちなみに，デフォルトではドライブ0，ドライブ1，ハードディスクの順に捜していくので，フロッピーディスクがささっていなければ自動的にハードディスクから立ち上がる。自分自身，それで困った経験はないし，それで困った人の話も聞いたことがない。よって，メモリスイッチをわざわざ書き換える必要は（特殊な事情があれば別として）ないのだ。

1)と3)については，次の第2部で述べよう。システム環境作りは楽しいのだ。

第2部 分割して統治せよ

真面目に本論である

さて，この第2部こそがハードディスクはどう使うとお得で賢いか，という話だ。

「分割して統治せよ」，といったのは誰だったか忘れたけれど，これは植民地政策だけでなく，ハードディスク政策にも非常に有効である。

たとえば，ハードディスクは使い込むと遅くなるのである（といっても普通の人なら1年くらい使ってやっと“あれ？”と思う程度だけれどね)。これは仕方がない。いろんなファイルを作ったり消したり変更したりしていると，だんだんハードディスクのなかはギタギタになってくるのだ。どういうことかというと，こういうことである。

ディスクに2Kバイトのファイル“ヘロロ”，200Kバイトの“デレレ”，500Kバイトの辞書，300Kバイトのワープロの順で入っていたとしよう。さて，あなたは2Kバイトのヘロロをエディットして，セーブした。そのとき，ヘロロは5Kバイトに膨らんでいた。しかし，ヘロロの元いた場所は2Kバイトしかない。じゃあ，どうやって格納されるかというと，残り3Kバイトはワープロの先のまだ使っていないところへ追いやられるのである。もちろん，ヘロロの2Kバイトのところには，残り3Kバイトがどこに入っているか書いてあるから，次に使うときはまたちゃんとつながって5Kバイトになる。

しかし，である。ハードディスクのヘッド君は2Kバイトを読んだあとで200+500

注4） 某電脳倶楽部などに「OPT.1キーを押しながら……」と書かれているのは，こういう意味があるのだ。

+300の約1Mバイト先まで行かないと全部は読めないのである。さあ，こういったことがたくさん（本当にたくさん起きないと，自覚症状はでないからそう心配することもないけれどね）起きるとどうなるかは考えなくてもわかる。ハードディスクのなかにはコマ切れになったファイルがたくさん詰まっていることになるのだ。

これを，ギタギタになったハードディスクと呼ぶ。いくらハードディスクが速いとはいえ，何10Mもの空間に点在したファイルをかき集めるにはそれなりの時間がかかるわけだから，ユーザーから見たら，「どうしてだろう。この頃，うちのハードディスクが遅くなった気がする」のだ。特に，頻繁に読み書きをするサブ辞書なんかが危ない気がする。見えないところで，ハードディスク君もたいへんなのだ。

これだけでも，分割して統治する理由になるけれど，ほかにもまだある。

ひとつのドライブにファイルがたくさんあると，探すのが面倒臭いのである。しかもDIRコマンドひとつでも，相手が何10Mもあると相当時間がかかる（これはディスクの残容量チェックに時間がかかっているのだ）。DIRを打ったらでれでれでれーっと100個以上のファイル名がスクロールなんかしたら，目的のファイルを捜すだけでもたいへんだ。サブディレクトリをいっぱい作ればいいかもしれないが，ディレクトリだってアッという間に2画面や3画面分はできてしまうし，階層構造を深くすると，結局探しにくくなるし，探すのに時間がかかる。WHEREコマンドでファイルをひとつ探すにも結構時間がかかる。ああ，ハードディスクという快適な環境を手に入れた瞬間，あなたは待つことが苦手になっていくのだ。

ほら，都市計画の必要性があると思うだろう。40Mどころか，10Mだって悩みがいがあるというものだ。

さて，このハードディスク都市計画には2つのレベルがある。

1）物理的に，どれだけの領域をいくつ確保するか。

2）どういった単位でサブディレクトリを作るか。

である。それぞれ，なかになにを入れるか，という問題は当然付いてまわる。で，基本的にHuman68k ver.2を中心に考えてみよう。ver.1しか持っていない人はごめんなさい。

基本は，ハードディスクにはまずアプリケーションを入れろということだ（これは常識）。アプリケーションならば書き換えられることもないからギタギタになることもないし，いつでもすぐ立ち上がるのが嬉しいし，仮にハードディスクがクラッシュしても（フロッピーディスクは落としても大丈夫だけれど，ハードディスクは落としたらきっと壊れる）マスターはフロッピーディスクにあるわけだから復旧作業がすぐできる，などなどである。

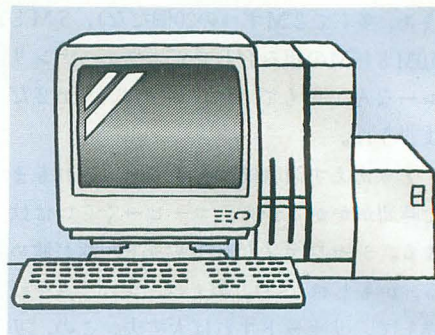
だから，ワープロや，エディタや，各種コマンドや，BASICや，持っている人はCなどもいい。専用のディレクトリを作って，フロッピーディスクからコピーしてやればいいから簡単だ。それから，ハードディスクに組み込めるアプリケーションのチェックも重要だ。

主なアプリケーションチェック

荻窪圭の独断による選択

神風やBusiness PROはハードディスクに組み込めるが，立ち上げ時に絶対プログラムディスクを挿入しなければならないというプロテクトがかかっている。困ったものだ。それから，こういったソフトにはハードディスクインストール用のバッチファイルが付いているものだが，それを鵜呑みにすると，苦労して作ったCONFIG.SYSやAUTOEXEC.BATまで書き換えられてしまうことがあるので注意だ。“見ず知らずのバッチファイルはなかを覗いてから実行する”のがいいね。よくできたバッチファイルは勉強にもなる。

CARD PROはそっくりそのままハードディスクに入った。ビジネス関係のソフトはやはり実用上ハードディスクが必要だから，当然といえば当然。

PrintShopにも，そっくりそのままハードディスクにインストールするバッチファイルが付いている。しかも，ハードディスク組み込み用のチェックプログラムがCONFIG.SYSを転送するかどうか聞いてくれるなど親切である。安心して身を任せられるハードディスクインストールだった。確かに，PrintShopはハードディスクでもないとやってらんないもんね。

MUSIC PROも組み込めるけれど，MIDI対応版になるとひとつ問題がある。プロテクトなのかなんなのか，“OPM”という名前のファイルがあるのだ。この名前は，OPMドライバが組み込まれているとデバイス名扱いになる。デバイス名はそのままではファイル名として使えない。よって，“OPMDRV”とMusic PRO [MIDI] は共存できないということになる。まあ，MIDI用の領域をひとつ作り，それを選択すると“OPMDRV”でなく“MIDIDRV”を組み込むようなシステムにすることで解決できることではある。

Zs'STAFFもぜひハードディスクに入れたいアプリケーションだ。プロテクトモジュールさえあれば，ハードディスクからだろうがなんだろうが問題なく動く。特に大容量のアウトラインフォントディスク（ver.2のみ）は，ハードディスクが必需といえよう。

これから出るアプリケーションには，きっとハードディスクユーザーに優しいものばかりになるだろう。また，ハードディスクに組み込むことにより，機動力が高まり（いつでもすぐ使える），ビジネスユーザー以外にも実用アプリケーションが楽しめるようになる。

あと，私が望むのはゲームだね。ゲームこそ余計な待ち時間なしに軽く楽しみたい。特に，ディスク3枚組以上の大作は，せめてデータだけでもハードディスクに組み込んで遊びたいものだ。ね。

領域確保したらファイルをコピー

適当に領域確保を行い，フロッピーディスクから立ち上げ直せば，Cからドライブ名がつく。最初に確保したところには一番

よく使うシステムやアプリケーションを入れたい。でかいアプリケーションがたくさん入るから，CONFIG.SYSやAUTOEXEC.BATなどよく書き換えるファイルを除いて，削除や変更はしないものを中心に組み込みたい。これを念頭に置いてリセットをし，ファイルのコピーを始めよう。

忘れてはならないのがディレクトリを見た時の美的配置。毎回DIRと打つ，あるいはビジュアルシェルを立ち上げるたびに見るわけだから，ディレクトリに混じって変なファイルがあるとみっともないし，デバイスドライバを集めたディレクトリは前のほうにほしいなど，システム関係は集めておきたい。これはファイルをコピーした順にディレクトリに配置されるので，慎重にやろう。そして，自分だけの贅沢なCONFIG.SYSやAUTOEXEC.BATを作る。

これが済んだら，次は辞書やデータをどうするかだ。プログラム開発用ドライブも快適なコンパイルのために用意したい。

たとえば，私は図4のようにしている。システムアプリケーション用13M，開発ツール用9M，データプログラム用7M，辞書専用1Mである。残り10Mは原野のまま予備である。いまのところこれで足りているからね。プログラム開発がメインの人はプログラム用にもっと確保したいだろうし，グラフィックをよく使う人は，グラフィックデータ用にもっと大きな領域を確保したいだろう。それでもって，何通りものシステムを使いたい人はそれなりの分割が必要だ。

そこで，不定期連載第1回である。

僕のCONFIG.SYS
君のAUTOEXEC.BAT

いきなりベテランというか，使いこなしている人の投稿である。東京都の長井清さんだ。ありがとうございます。

さっそく図5と6である。X68000は初代機種プラス1Mの増設RAM，20Mのハードディスクだそうだ。CONFIG.SYSをお見受けしたところ，Human68kはver.1。だから，ハードディスクはすべてAドライブとして使っているようだ。しかも，コプロを装備している。AUTOEXEC.BATを見ると，X68000をその人がどんな用途で使っているかわかる。長井さんは誰が見たってプログラム開発だ。しかも，使っているエディタはWINDEX(まさか，環境変数だけセットして，その実microEmacsを使っているとかはないよね)。

では，きれいにまとまったこの2つを簡単に解説してみよう。

CONFIG.SYSは別に変なことはしていない。プリンタスプール用に32Kバイトのバッファが確保してあって，出力はプリンタの内部フォントを使っていること，コプロ用の浮動小数点ドライバFLOAT3+(そう，長井さんは昨年8月号で発表された3+の作者なのです)を使っていること，RAMディスクに640Kバイト確保していることなどだ。特徴は，コマンドシェルのオプションと見た。バッチファイル用エリアになんと14×512の7Kバイト，環境エリアに6×512の3Kバイトも取っているのだ。なかなかその筋といえよう。バッチファイルのプロだ。FILES＝30やらBUFFER＝50もメインメモリを思う存分快適な環境のために使えるハードディスクのおいしさを主張している。

AUTOEXEC.BATもその筋だ。それは6行目の“OPMDRV OFF”に現れている。実行スピード確保のためにOPMドライバをOFFしているのだ。さすが，開発環境の追求が見られる。プロンプトもなかなか凝っていて楽しい。プロンプトはその人のセンスや趣味が出るからなあ。皆さんも，いつまでも“A>”に甘んじていないよう

図4　私のCONFIG.SYS

```
FILES     = 20
BUFFERS   = 60 1024
LASTDRIVE = Z:
KEY       = ¥CNF¥KEY.SYS
USKCG     = ¥CNF¥USKCG.SYS
BELL      = ¥BEEP¥WALK_R.PCM
DEVICE    = ¥SYS¥PRNDRV.SYS #/M1 /B24
DEVICE    = ¥SYS¥RAMDISK.SYS #M512
DEVICE    = ¥SYS¥PCMDRV.SYS
DEVICE    = ¥SYS¥ASK68K.SYS F:¥X68K_M.DIC F:¥X68K_S.DIC ¥ASK¥OGIKUBO.ASK
DEVICE    = ¥SYS¥OPMDRV.X
DEVICE    = ¥SYS¥FLOAT2+.X
DEVICE    = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4 /O /VC:¥HIS¥
PROGRAM   = ¥BIN¥DRIVE A: E:
PROGRAM   = ¥BIN¥DRIVE B: F:
PROGRAM   = ¥BIN¥DRIVE C: E:
PROGRAM   = ¥BIN¥DRIVE D: F:
PROCESS   = 2 2 10
BREAK     = ON
VERIFY    = OFF
```

図5　長井さんのCONFIG.SYS

```
FILES   = 30
BUFFERS = 50
BELL    = ¥SYS¥BEEP.SYS
DEVICE  = ¥sys¥PRNDRV3.SYS #/B32 /M1
DEVICE  = ¥SYS¥PCMDRV.SYS
DEVICE  = ¥SYS¥OPMDRV.X
DEVICE  = ¥SYS¥ASK68K.SYS A:¥DIC¥X68K_M.DIC A:¥DIC¥X68K_S.DIC
DEVICE  = ¥SYS¥FLOAT3+.X
DEVICE  = ¥SYS¥RAMDISK.SYS #M640
VERIFY  = ON
SHELL   = COMMAND.X /B:14 /E:6
```

図6　長井さんのAUTOEXEC.BAT

```
PATH A:¥;A:¥BIN;A:¥CC;A:¥BC;A:¥BASIC2;A:¥ETC;A:¥BAT
COPY A:¥OPMDATA¥SCEND.OPM OPM
SET TEMP=D:
set lib=A:¥LIB
set include=A:¥INCLUDE
set windex=a:¥windex
a:¥sys¥opmdrv.off
prompt $E[60C$E[35m$T$E[5C$D$_$E[0m$P$S$G
```

に。ちなみに，これは，シアンの強調色で画面右に日付と時間が出力され，次の行頭に白でカレントディレクトリと“>”が出るようにしてある。日付と時間を右に寄せるところが憎いね。こうすると邪魔じゃない。私としては，“PS$G”を黄色なんかにしたいものだ。

第1回目にしてはいささか地味だったが，これはHuman68k ver.1にハードディスクという装備がオーソドックスで綺麗なシステム環境を招いたのだといえる。ただ，私と趣味の合わないところが1点ある。ハードディスクをAドライブで使っているところだ。私は“ハードディスクはCドライブから”派である。だから，CONFIG.SYSにPROGRAMコマンドを4つ並べて，A～DになるはずのハードディスクをC～Fにシフトしている（もちろんA，Bはフロッピーディスク)。これはもう，習慣の問題であるから，どっちがいいとはいえない。

引き続き，フロッピーベース，ハードディスクベースを問わず，CONFIG.SYSやAUTOEXEC.BATなど「僕のシステム環境コーナー」は，皆様からの投稿を待っている。アクロバティックなものでもアブないものでも結構。

第3部
バックアップから活用へ

バックアップあれこれ

さて，ハードディスクを使っている人に尋ねてみたい。あなたは，何回ハードディスクのバックアップを取ったことがありますか？　どのパソコン誌でも単行本でもハードディスクといえばこの問題が欠かせない。金がからむビジネスユーザーには当然として，あなた，バックアップを何回取りましたか？

バックアップといっても，X68000ユーザーにハードディスクバックアップ用ストリーマなんて用意している人は多分いないだろうから，フロッピーディスクを何枚も用意しているのだろう。

これがまた面倒臭い。ついでに，どのコマンドでバックアップするかも重要だ。“BACKUP.X”？　いやいや，それではまだまだ甘い。BACKUP.Xはハードディスクの内容をそのままひとつの専用ファイルに落としていくコマンドなのだ。試しにやってみると，ディレクトリひとつをフロッピーディスクに退避すると，なかにどれだけファイルが入っていようとも，“BACKUP01.FIL”なるひとつのファイルになってしまうことがわかる。どういうことかというと，RESTORE.Xを使ってハードディスクに戻すことがない限り，なんの役にも立たないファイルなのだ。そんなディスケットが何枚あってもなんか嬉しくない。

では，“COPY2.X”？　これも，ハードディスク専用のCOPYコマンドだが，いまひとつ気に入らない。コマンド一発で全部コピーというわけにはいかないからだ。ディレクトリごとにやらねばならない。なおかつ，そのままコピーするわけではない，ときたもんだ。たとえば，実行ファイル（うんちゃら.X)をフロッピードライブにCOPY2で退避してみるといい。ファイル名は同じだけれど，なぜかファイルの大きさが違う。実行してみる。できない。

そこで，“COPYALL.X”の登場である。

知る人ぞ知る“COPYALL.X”は非常に便利なコマンドである。便利ということは機能がいろいろあるということで，機能がいろいろあるということは，実行時に付けるオプションが多いということで，ということは使い方を覚えるのが面倒だということで，つい敬遠されてしまう，となる。

しかし，これは便利なのである。コピー先にないディレクトリは作ってからそこに

Cプログラマはここまで追求する

「君のCONFIG.SYS，僕のAUTOEXEC.BAT」を送っていただいた長井氏が，「ハードディスクユーザーに贈る，RAMディスクをXCで上手に使うバッチファイル」というのをオマケで送ってくれたので，ここで紹介することにしよう（図7)。

ただ，単にXCコンパイラをRAMディスクに転送するだけのものだが，実に無駄なく，しかも環境変数まで書き換えてくれるのだ。いくつかここで使われている技を見てみることにしよう。

1）オプション付き起動

“−B”か“/B”オプションでBASIC to Cコンバータ（ラベルCPYBC)，“−Y”か“/Y”オプションでDOSLIB.AとIOCSLIB.Aを，“−W”か“/W”オプションでBASLIB.Aを転送してくれるのだ。さすがにこれらのオプション3つ全部一度に，というわけにはいかないけど。

2）画面になにも出さない

すべてのコマンドが“>NUL”，つまりメッセージの出力先をダミーにしているので，ユーザーはなにをやっているかわからないという寸法。私なんぞは自分で使うバッチファイルなのだから，実行するコマンドは全部自分の目で見て確認したい性分なのだ。いつなにが起きるかわからないしね。

3）setコマンドでPATHを追加する

これは「set PATH=D:¥XC;%PATH%」の部分がミソ。ただのPATHコマンドだと前に設定したPATHが消えてしまうけど，こうすれば大丈夫。甘露な技である。

4）SHIFTを使いこなせ

巧妙に配置されたSHIFTコマンドにお気づきだろうか。SHIFTコマンドによった複数オプションの記述を可能にしているのだ。ひとつ目のオプションの処理が終わると，SHIFTによって2つ目のオプションが“%1”に入り，再びその処理が行われる。バッチファイルの奥の深さのひとつだろう。

というわけで，よくできたバッチファイルである。RAMディスクにCC.Xがあればオプションの処理に行ってしまうところなど，常道とはいえ，見ていて気持ちいい。ただ，setコマンドを最初に行っているのが気になる。些細なことだが，COPYが失敗することを考慮し，setは後ろに持ってきてもよかったのでは？

いずれにしても，皆さんもこれを参考に精進してほしい。

図7

```
echo off
rem  C Complier  を  RAM disk  に転送する
if EXIST D:¥XC¥CC.X goto ETC_COPY
md      D:¥XC > NUL
set     cc=D:¥XC
set     PATH=D:¥XC;%PATH%;
set     lib=D:¥XC
copy    A:¥COMMAND.X     D:¥      >NUL
copy    A:¥BIN¥CC.X      D:¥XC    >NUL
copy    A:¥BIN¥CC.X      D:¥XC    >NUL
copy    A:¥BIN¥AS.X      D:¥XC    >NUL
copy    A:¥BIN¥LK.X      D:¥XC    >NUL
copy    A:¥BIN¥CASH.X    D:¥XC    >NUL
copy    A:¥CC¥*.X        D:¥XC    >NUL
copy    A:¥LIB¥CLIB.X    D:¥XC    >NUL

:ETC_COPY
if "%1" == "-B" goto CPYBC
if "%1" == "/B" goto CPYBC
if "%1" == "-Y" goto CPYLIB
if "%1" == "/Y" goto CPYLIB
if "%1" == "-W" goto CPYBAS
if "%1" == "/W" goto CPYBAS

:CPYLIB
if EXIST D:¥XC¥IOCSLIB.A goto CPY2
copy A:¥LIB¥IOCSLIB.A   D:¥XC   >NUL
:CPY2
if EXIST D:¥XC¥DOSLIB.A  goto CPY3
copy A:¥LIB¥IOCSLIB.A   D:¥XC   >NUL
:CPY3
SHIFT
goto ETC_COPY

:CPYBC
if EXIST D:¥XC¥BC.X goto CPY4
set      bc=D:¥XC
copy A:¥BC¥*.*          D:¥XC   >NUL
:CPY4

:CPYBAS
if EXIST D:¥XC¥BASLIB.A goto CPY5
copy A:¥LIB¥BASLIB.A    D:¥XC   >NUL
:CPY5
SHIFT
goto ETC_COPY

:END
```

コピーしてくれるわ，コピー先のディスケットがいっぱいになったらメッセージを出してディスクの入れ替えを促してくれるわ（当然，ひとつのファイルが2枚のディスクにまたがるといったアコギなことはない），コピー元とコピー先を比較して日付の新しいものだけをコピーしてくれるわの優れものなのだ。

で，COPYALLの最大の長所は，BACKUP.XやCOPY2.Xと違って，コピーしたファイルを好きに扱えることである。つまり，必要なものだけをハードディスクに戻せるのである。

これは，ハードディスクが壊れたときのためのバックアップだけでなく，ハードディスクの領域確保をやり直したいとき（都市計画の見直しね）などに便利なのだ。

また，使いすぎてギタギタになったハードディスクを綺麗に整理するときにも使える。一回退避させて，ハードディスクの中身を全部DELし，戻す。この作業がときには無秩序化したハードディスクのリフレッシュにもなるのである。

究極の4タイプ

しかしまあ，何10Mバイトのデータをフロッピーディスクに放り込もうというのだから，何10枚のフロッピーディスクを用意して，何10回ものディスクの抜き差しを経験せねばならないのは非常に面倒臭い。

では，どうするか。

1）壊れたときは壊れたときさ，この世に壊れないものなんてねえや！　と悟る。ちなみに，私はこのタイプである。くわばらくわばら。

2）バックアップは面倒だけど，大事なデータが壊れるのはいやだ，と，ハードディスクにはアプリケーションしか入れないようにする。大事なデータはフロッピーだ。ついでに，ハードディスクに入っているファイルの名前くらいはTREE.Xの/Fオプションでプリントアウトくらいはしておく。CONFIG.SYSとAUTOEXEC.BATも忘れずに。これで，壊れても復元は可能だ。

3）せっかくだから，データもハードディスクに入れるけれど，COPYALLの新しいファイルだけコピーするオプションでデータだけはバックアップを取っておく。このパターンが一番無難かな？

4）律儀にせっせとバックアップを取る。おお，マメなやつ。

まあ，これは，人それぞれだけれど，くれぐれも「ハードディスクがクラッシュして大事なデータが壊れちまったじゃねえか。どうしてくれるんだよう！」と，メーカーに怒ったりはしないように。壊れないハードディスクはこの世には存在しないのである。もっとも，メーカーが悪い場合もあるから，理不尽でない文句はいっぱいいってやろう。

ただ，ハードディスクがクラッシュといっても，何10M全部が一気に壊れることはまずない。どこかのセクタ（おう，久々に難しい言葉を使ってしまった）だけが駄目になったり，どこかのトラックだけが駄目になったりするあたりが一番ありそうだ。そんなときは，まず大丈夫なやつを全部バックアップし，それから復元(RECOVER)に挑戦してみよう。うまくいくときはうまくいくぞ。

管理領域が壊れたときは悲惨だけどね。ファイルはあるけれど，そのファイルがどこにあるかしまっておくところがパーになったのだから。

また，アクセスランプだけが壊れたとか，基盤の接触が1カ所だけ悪くなったとか，電源部がいかれたなどというディスクはなんとか大丈夫だということもあるから，そういったときは，ほっと安堵の息をつこう。

HDで複数のシステムを使う

最後になってしまったが，複数のシステムでハードディスクを使うことを考える。パターンは2つしかない。

1）いうまでもなくOS-9との共存。

2）Human68kの環境を2種類載せる。

である。前者から見ていこう。これはもうただひとつだけ注意すればいい（そのあと怒ってもいいよ）。OS-9/68000をハードディスクに組み込むときは，“最初の領域に放り込まねばならない”ようになっているのである。うーん，わかりにくい表現だ。逆にいうと，たとえばHuman68kですでに40Mバイトのうち20Mバイトを使っているとき，残りの20MにOS-9を入れようと思っても駄目だ，ということだ。そういうときは一回全部領域を解放して，まっさらにしてから，まずOS-9を組み込み，残りにHuman68kを組み込むという形にしなければならないのである。理由はよく知らないけれど，そうなのだ。

で，ハードディスクにシステムを複数組み込むと，起動時に選択画面が出てくる。出てこないときは，“自動実行”属性の領域を持っているときだ。そんなときでも，HELPキーを押しながら立ち上げると，メニューが無事に出てくる。

こんなわけで，ひとつのハードディスクに複数のシステムを組み込んで，用途によって好きなほうで立ち上げるといったこともできるのであった。

*　　　*

とにかくハードディスクは便利だ。そのメリットはファイルがたくさん入ってディスクの入れ替えが少なくなることだけでなく，そのアクセスの速さから“辞書をわざわざRAMディスクに持たなくてもよい”とか，“RAMディスクを高速コンパイルのためにCなどの開発用やCのコンパイラディスクとして使う必要がない”現象（もっとも，Cを頻繁に使う人は“ハードディスクでも遅い。RAMが何Mも欲しい！”そうだ)が出てくる。すると，RAMディスクをたくさん確保しなくてもよい→メインメモリがたくさん取れる→大きなプログラムを開発したり，チャイルドプロセスでいくつもプログラムを起動したりできる，といったおいしい流れが確保される。これは実感。メインメモリの増設はまあ，1Mくらいしたいけれども，それでかなり快適な生活が送れる。これから，おいしい常駐プログラムやバックグラウンドプログラムがどんどん増えるだろうが，メインメモリがたくさんあるおかげでメモリの残りを気にしないで組み込めてしまう。

これが肝心だね。早く来い来い，StationeryPRO-68Kよ（注5）。

と，いうわけだ。はあ，長かった。安くなったといえども，10万円くらいはするけれど，Human68kでよく遊ぶ人ならば私と一緒にクレジットカード地獄で徳を磨こうではないか。

注5）いきなり，なんの噂もないままにシャープの広告に現れてビックリした私。

PRINTER

BASICで書けるハードコピープログラム

超初歩的硬式複写術入門

Mounai Toshiyuki
毛内 俊行

プリンタの記事で必ず現れるのがハードコピープログラム。ここでは，そのもっとも基本となる部分をBASICで作ってみましょう。これなら拡張も簡単，しかもコンパイルすればちゃんと使える速度になるのです。

白と黒だけで多色表現

「プリンタでフルカラーのハードコピーを行うにはどうすればいいのですか？」という質問に対する反問，「どうやったらできると思いますか？」

＊　＊　＊

最近のパソコンは，平気な顔をしてグラフィック画面に何千，何万色もの色を表示することができたりする。X68000の場合は，最大65536色の同時発色が可能である。ところが，そのグラフィック画面のハードコピーを取ろうとしても，プリンタは白と黒の2色しか表示できない場合がほとんどである。これでは，表情豊かなグラフィックに対応しきれるわけがない。

それではモノクロでは多色グラフィックを表現できないのかというと，そんなことはない。新聞の写真なんかを見ればわかるとおり，白と黒の2色だけで立派に写真を表現している。新聞の写真は，白い紙の上に大きさの異なる黒い小さな点を配置することによって，明るさを表現している。つまり写真の明るいところは小さな点，暗いところは大きな点を置けば，明るさの明暗だけで絵を表現できるわけだ。この方法は，そのままプリンタでも使えるわけではないが，モノクロでの色調表現法を示唆してくれている。

プリンタとディスプレイの違いを挙げてみよう。ディスプレイは色数が多い。プリンタはドットが細かい。それならば，ディスプレイ上の1ドットに対して，プリンタの数ドットを対応させてやれば，タイリングペイントの要領で多色表示ができるだろう。

一度に複数の色を指定するわけにはいかないから，色コードをRGBの要素に分解し，黄色・シアン・マゼンタのインクリボンに分けて3回重ね打ちすればカラー表現ができそうだ。そのドットに対応するRGBの比率が同じ確率になるようにプリンタに対応させるわけだ。ドットの割り振り，色の決定法などに違いはあっても，この考え方がカラーハードコピープログラムの基本となる。

「えー，本当にそんなうざったいことをやってるんですか？」「ほかにどうしろというんですか」

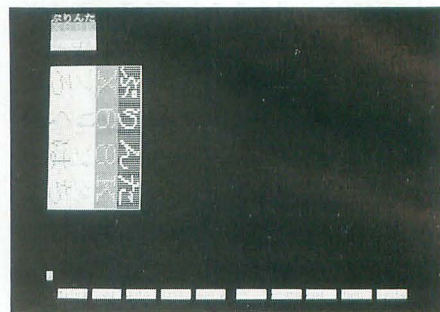
DO IT YOURSELF

新しく48ピンのカラープリンタが発表された。当然，48ピン対応のハードコピープログラムを求める人もいるわけだ。同様に(たぶん出ないと思うが)，64ピンプリンタとかが現れれば，また新しいプログラムを必要とする人が出てくる。これでは新しいプリンタが出るたびにハードコピープログラムを作らなくてはならないのだが，既製のものを使うよりも，自分で作れればそれがいちばんいい。

ハードコピープログラムのもとになるビットイメージ印字は，たいていのプリンタでその機能をもっとも発揮できる分野だ。それを使いこなすことは，ある意味でそのプリンタを究めた証にもなるだろう。プリンタを買ったら，ハードコピープログラムくらい作れずにどうする，ということで，ここではもっとも基本となるモノクロハードコピーを実現することにしよう。

「どうやって開発するのですか？」「BASICを使います」

＊　＊　＊

プリンタの制御はすべて，コントロールコード，エスケープシーケンスを送ることで行われる。BASICからでも扱える，むしろBASICがもっともプリンタを扱いやすいシステムかもしれない。

ふつうBASICでやると恐ろしく遅いものができあがってしまうが，そこはそれ，XCをコンパイラとして使ってやれば実用に耐える速度になってくれる。

X68000の場合，X-BASICのLPRINT文では出力できないコードがあるので，プリンタの制御はもっぱらファイルを使って行われる。PRN，LPTなどのプリンタを表すシステム予約ファイルを使用するのである。このうち，PRNは主に文字出力に使うことを目的としたもので漢字IN/OUTコードを出力するため直接のプリンタ制御には適していない。プリンタへのコントロールコードはLPTというファイルを介して転送することになる。

余談だが，同様にFM音源関数がなくてもファイル名OPMを開いてデータを送れば演奏は可能であるし，PCM音源用の関数の代わりにファイル名PCMを使うこともでき，AUXではRS-232C制御をすることも不可能ではない。ただし，これらのものとBASICレベルの関数などを混用したものをコンパイルした場合，システムワークエリアなどの使用状況がダブってしまい，誤動作が起きることがあるかもしれない。Cコンパイラのマニュアルでできるだけ同レベルの関数を使うようにと注意されている以上，乱用は避けたい。

ややこしいプリンタの構造

プリンタのマニュアルを1冊読もうとすると，死ぬほど根気がいる。今回主に使うコントロールコードは，16ドットイメージ印字というやつだ。

プリンタも8ピン，16ピン，24ピン，48ピン……とどんどんピン数が増えていく傾向にあるが，これも処理する単位が変わってくるだけで，やらなければならないことは変わらない。むしろ，ピン数が少ないほうが考慮しなくてはならないことが増えてくる。突き詰めていけば，1988年11月号で栗野雅彦氏が行った1ピンハードコピーとなる。すでにそういったアプローチについての解説も行われているので，拡張は皆さんにまかせたい。ここではあくまでも最低限ということで16ピンを採用する。

この16ドットイメージ印字というコードは，データを送ると縦に16ドットのパターンを印字する。データは2進数のビットイメージで表現されるので，人間はまず，16ドット＝2バイト分のデータをビット単位で考えてやらなくてはならない。まずこれがややこしい。

さらに，これらのビットイメージが縦に並んでいる。X68000を除くほとんどのマシンのVRAMの構造が，1バイトで8ドットを表現しているのを知っている人も多いと思うが，これは横方向に並んでいる。つまり，ドットの列の方向が，CRT画面とプリンタでは90°違うのである。もし，マシン語でハードコピープログラムを作ろうとしたら，この縦横の変換をしてやらなくてはならない。たいていの人は，この縦横の変換をしようとして，頭の回線がショートする。プリンタでグラフィックを使おうとする人はまず，このへんで挫折する。

今回作成するハードコピープログラムは，X-BASICを使って作られている。BASICを使うとグラフィック画面の考え方がドット単位になるので，一見難しそうなプログラムのテクニックを理解するのには好都合というわけだ。また，誰にでも簡単に理解できるように，難しいテクニックなどは一切使っていない。もっとも，誰にでも理解できるといえば聞こえはいいが，実は私もハードコピープログラムについては素人同然なので，難しい話は一切できないのが本当だ。だからいまからびびる必要はまるでない。

それでは実際に，どうやってプリンタで明暗をつけているのだろうか？　ひとつのドットでは大きさを変えることはできない。それなら複数のドットを用いて明るさを表現してやればよい。今回は8ドット（4×2ドット）で，点ひとつを表現している。

こんなに大きな点で，ちゃんと絵が描けるのかと，心配するかもしれないが，プリンタのドットそのものが結構小さいので，気にすることはない。ただし，さすがにこれだけ大きいドットをこしらえるとプリンタの表示範囲が狭くなる。困ったことに80桁表示のプリンタでは，画面サイズが256×256のときでなくては全画面のコピーが取れない。だから，それ以上の画面サイズでコピーを取りたい人は，自分で工夫してほしい。また，プリンタの表示範囲を有効に使うため，プリンタに出力される絵は90°回転している。つまり，絵は横に寝た格好になっている。

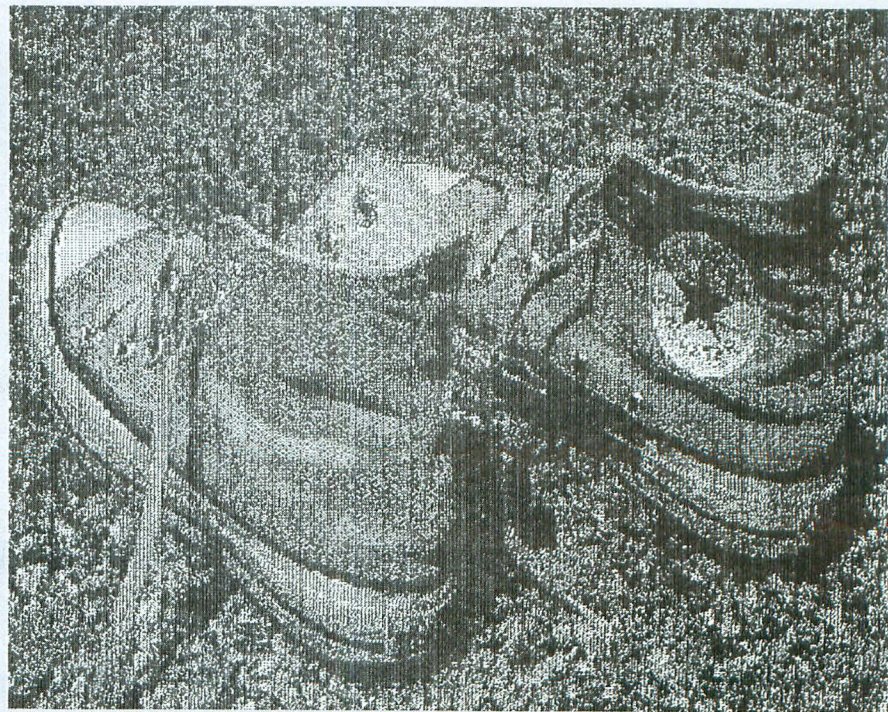
さて実際の表示だが，ここでは明るさを点の大きさで表す代わりに，色をドットのパターンに置き換えている。これはタイルチェンジとも呼ばれ，昔からよく使われている方法である。本当は，カラーコードをh，s，vの3つに分解して，明るさだけで表示しようと思ったが，分解する方法がよくわからないので，結局あきらめてしまった。

だからパターンの作成は，カラーコードのみで行っている。聞いたところによると，このパターンにも「□×△氏の作った○○のパターン」なんていうのがいくつかあるらしいのだけど，私はそんなものをまったく知らないので，カラーコードを適当な数で割っただけといういいかげんなものにしてしまった。

しかしこれでも，結構見られるようなパターンになっているから世の中不思議である。このパターンを書き換えたい人は，リスト1の400行以降を書き換えてみるといい。詳しい内容は，このあとのプログラムの解説で説明する。

プログラムの使い方

プログラムはリスト1，リスト2の2つが用意されている。このうち，ハードコピーに必要なのはリスト1である。あとで詳しく述べるが，リスト2はプリンタへのデータを画面へ出力するためのサブプログラムである。プリンタを持っていない人はあらかじめリスト2を入力しておいて，リス

図　印字例

ト１の1000行以降をリスト２と取り換えておくといいだろう。

プログラムの実行方法はBASICからRUNするだけでいい。するといきなり，

image file name=■

と表示してキー入力待ちになる。これはグラフィックのイメージファイルをディスクから読み込むためのもので，ファイル名を入力してリターンキーを押せばいい。あとは自動的にハードコピーを取ってくれる。あらかじめ画面に描かれている絵のコピーを取りたいときには，130行を削除しておくこと。

リスト２を使うときは，リスト１の1000行以降とリスト２を置き換えて使う。リスト１の1000行以降は，プリンタの制御を行う関数が集まっている。リスト２は，その関数をそのまま画面出力用に使っているので，置き換えが可能なのである。リスト２をつけた状態でRUNした場合，自動的にサンプルの絵を描いて，そのまま画面に白黒のコピーを取ってくれる。だからグラフィックデータがディスクになくても関係なく，画面にコピーを取ることができる。

プログラムの解説

先ほども述べたとおり，リストは２つ用意してある。リスト１はメインプログラムで，ハードコピープログラムの本体である。プリンタでハードコピーを取るならとりあえずこれだけでいいだろう。リスト２は，プリンタへの出力をCRT画面に送るためのプログラムで，実際にはプリンタの16ドットイメージ印字というやつを，画面でエミュレートしている。使い方は何度も述べたように，リスト１の行番号1000以降をリスト２に取り換えるだけ。実行すれば，自動的に画面にイメージ表示を行う。

実際にプログラムを見てみよう。280行で，関数dotsetに８つのデータを送っている。この関数が，プリンタにドットを印字させる関数で，これらのデータはいずれも４ビットだ。これは１ドット分のデータを縦４ビットで表現するためである。関数内ではこのデータを，プリンタのヘッドに合わせて８ビットのデータに変換している。

400行からは先ほども書いたように，マトリクスパターンの作成である。作成されたパターンはp1，p2というグローバル変数に代入される。p1はカラーコードを4096で割っただけ，p2はカラーコードと4095の論理積（and）を取り，その結果を256で割ったものだ。プリンタ上では，この２つのデータを横に並べることによって１ドットを表現している。

なお，p1，p2の２つの変数を15で排他的論理和（xor）を取っている。これは，CRTとプリンタではドットの白黒が逆になるので，あらかじめビットを反転させているからである。このパターン変換は結構いいかげんなので，やはり少し出力が汚い。もし力がある人なら，このあたりを自分なりに書き換えてみてもいいかもしれない。さらにそれ以上の出力を追求する人には1988年９月号の丹明彦氏によるディザ法の解説，11月号の栗野雅彦氏による階調表現のアルゴリズムなどを参考にしてほしい。

1000行以降は，プリンタへの出力である。変数xmaxおよびymaxを決定することによって，画面サイズを決定している。一応，最大値は256×256となっているが，それ以上でもプリンタに印字できれば問題なく使えるはずである。

なお，このプログラムはシャープのCZ系プリンタに対応しているため，エプソンやNEC系列のプリンタを使っている人はコントロールコードなどの書き換えが必要である。マニュアルをよく見て書き換えること。書き換えが必要なのは1050行と，1130行の２行だけ。1050行は改行幅を決定するためのコードで，1130行は16ビットイメージモードのコードを作成している。ただし，NEC系列のプリンタでは，ビットイメージモードのデータがビット単位で上下逆であるので，さらに，以下の２行の変更を行ってほしい。

```
1190 wd1(0)=d1+d2*16:wd1(1)=d3+d4*16
1200 wd2(0)=d5+d6*16:wd2(1)=d7+d8*16
```

また，エプソン系列のプリンタ（ESC/P対応）を使っている人は，ビットイメージモードに16ドットモードがないので，24ドットモードを使って下位16ドットだけで印

リスト１

```
 10 /*-----------------------------------------
 20 /*     ハードコピーのプログラムです
 30 /*   Programed by Toshiyuki Mounai 1989
 40 /*-----------------------------------------
 50 str esc,lf,cr
 60 str img,pit
 70 int xmax,ymax
 80 int x,y,col,p1,p2,px,py
 90 int b1,b2,b3,b4,b5,b6,b7,b8
100 esc=chr$(&H1B)
110 lf=chr$(10)
120 cr=chr$(13)
130 glod()
140 p_opn()
150 x=0
160 repeat
170   imagemd(ymax)
180   y=ymax-1
190   repeat
200     pindat(point(x,y))
210     b1=p1:b5=p2
220     pindat(point(x+1,y))
230     b2=p1:b6=p2
240     pindat(point(x+2,y))
250     b3=p1:b7=p2
260     pindat(point(x+3,y))
270     b4=p1:b8=p2
280     dotset(b1,b2,b3,b4,b5,b6,b7,b8)
290     y=y-1
300   until y<0
310   crlf()
320   x=x+4
330 until x>=xmax
340 p_close()
350 end
360 /*-----------------------------------------
370 /* マトリクスパターン作成  in :(col)
380 /*                          out:p1,p2
390 /*-----------------------------------------
400 func pindat(col)
410 p1=col/4096
420 p1=p1 xor 15
430 p2=col and 4095
440 p2=p2/256
```

字するか，データのセット部分から24ドット対応に改造してほしい。おそらく問題はないと思う。もちろん，16ドットだけ使う場合は改行幅も16ドット分に変えるのを忘れずに。

リスト2のCRT出力では，p_closeなどのように，なにも作業をしていない関数もある。これは，プリンタ出力用と互換性を持たせるために設けたものなので，削除しないでほしい。

また，リスト2では1030行からの関数p_opnに，サンプル用のグラフィックを書くためのプログラムが入っているので，ここは適当に書き換えて好きな絵を表示してみるのもいいだろう。

いくつかの注意

プログラムの中身はとても簡単にまとめたつもりである。プログラムの長さも手頃だし，入力して自分の好きなように改造するのにはちょうどいいだろう。

なお，このプログラムでハードコピーを取るときは，できればコンパイラでコンパイルしてから使ってほしい。コンパイルしないと，256×256の絵をハードコピーするのに，おそらく30分近くかかってしまうだろう。

またプリンタというのは，なかなかいいかげんな機械で，改行幅なんかは最終的に自分で管理しなくてはならない。私が使っているNEC系列のプリンタでは，同じn/120インチの改行幅指定をしても，改行幅がCZ系のプリンタと異なってしまった。

そろそろ終わり

と，いうわけで初めて作ったわりにはまともに動いたプログラムなのだが，やはりマトリクスパターンの作成を適当にごまかしたせいか，少しパターンが汚いような気がする。このへんはやはり気合を入れてやらなくてはいけないところなのだろう。

なお，誤解されると困るのだが，これは完成されたプログラムではなく，改良されるべきプログラムである。改良できるところは山ほどあるので各自のアイデアで立派なハードコピープログラムに仕上げてほしい。

```
 450 p2=p2 xor 15
 460 endfunc
1000 /*-----------------------------------------
1010 /* プリンタ制御 (CZ対応)
1020 /*-----------------------------------------
1030 func p_opn()
1040 xmax=256:ymax=256      /* 画面サイズ指定(最大256×256)
1050 pit=esc+"%9"+chr$(15) /* 改行幅指定(n/120インチ,CZ)
1060 fn=fopen("lpt","w")
1070 fwrites(pit,fn)
1080 endfunc
1090 /*
1100 func imagemd(lng)
1110 str slg
1120 slg=right$("000"+str$(lng*2),4)
1130 img=esc+"I"+slg     /* ビットイメージモード(CZ)
1140 fwrites(img,fn)
1150 endfunc
1160 /*
1170 func dotset(d1,d2,d3,d4,d5,d6,d7,d8)
1180 dim char wd1(1),wd2(1)
1190 wd1(0)=d1*16+d2:wd1(1)=d3*16+d4
1200 wd2(0)=d5*16+d6:wd2(1)=d7*16+d8
1210 fwrite(wd1,2,fn)
1220 fwrite(wd2,2,fn)
1230 endfunc
1240 /*
1250 func crlf()
1260 fwrites(cr,fn)
1270 fwrites(lf,fn)
1280 endfunc
1230 /*
1300 func p_close()
1310 fclose(fn)
1320 endfunc
1330 /*
1340 func glod()
1350 str ifname
1360 screen 1,3,1,1
1370 input "image file name=",ifname
1380 cls
1390 img_load(ifname,0,0)
1400 endfunc
```

リスト2

```
1000 /*-----------------------------------------
1010 /*    CRT表示デモンストレーション
1020 /*-----------------------------------------
1030 func p_opn()
1040 xmax=64:ymax=64
1050 px=0:py=xmax+30
1060 screen 1,3,1,1
1070 fill(0,0,xmax,15,32769)
1080 fill(0,16,xmax,31,33442)
1090 fill(0,32,xmax,47,49095)
1100 fill(0,48,xmax,64,63220)
1110 symbol(0,0,"ぷりんた",1,1,1,65293,0)
1120 symbol(0,16,"X68K",1,1,1,58390,0)
1130 symbol(0,32,"プリンタ",1,1,1,17763,0)
1140 symbol(0,48,"ろくはち",1,1,1,10696,0)
1150 fill(px,py,px+ymax*2,py+xmax*4,65535)
1160 endfunc
1170 /*
1180 func imagemd(lng)
1190 endfunc
1200 /*
1210 func dotset(d1,d2,d3,d4,d5,d6,d7,d8)
1220 int wd1,wd2
1230 wd1=d1*4096+d2*256+d3*16+d4
1240 wd2=d5*4096+d6*256+d7*16+d8
1250 line(px,py,px,py+15,0,wd1)
1260 line(px+1,py,px+1,py+15,0,wd2)
1270 px=px+2
1280 endfunc
1290 /*
1300 func crlf()
1310 px=0
1320 py=py+16
1330 endfunc
1340 /*
1350 func p_close()
1360 endfunc
1370 /*
1380 func glod()
1390 endfunc
```

PRINTER

プリンタバッファクリア機能付き

COPYキーメニュー

Miyajima Yasushi
宮島 靖

付いていることを知っていても，なかなか使われないHuman68kのプリンタスプーラ機能。そこでこのプログラムの登場です。メニューで選んでバッファクリアできるようにしました。COPYキーの割り込みを使った常駐プログラムです。

プリンタスプーラってなに?

さて，皆さんはプリンタスプーラというものをご存じでしょうか? プリンタで何百行，何千行もあるものを印刷するとなると，高速ドットインパクト式のプリンタでもそうとう時間がかかってしまいます。

いくらプリンタが速くなったといっても，せいぜい1秒間に100文字打つのが精一杯。それに対して，10MHzのCPUは1秒間に何十万字ものデータを転送することができます。プリンタのような機械動作のデバイスはCPUの動作速度に比べてはるかにトロいわけです。

プリンタはデータを受け取って処理するまでの間，CPUにBUSY信号を送ります。CPUはじっとプリンタを見張っていて，BUSYでなくなったら次のデータを送るという動作を繰り返します。CPUは印刷に夢中になっているほとんどの時間，実はプリンタに次の文字を送れるようになるまでじっと待っているだけなのです。

これでは困ったちゃんなので，考え出されたのが，プリンタスプーラです。プリントアウトすべきデータをプリンタに渡すフリをしてバッファに溜め込んでおき，普段は印刷以外の仕事をしていて，たま～に(といっても1/60秒ごととかそのくらい) 割り込みをかけて，バッファに印刷すべきものがあるかどうかを調べ，あったら1文字プリンタに出力するというような方法です。この方法を使うとCPUを効率よく使えるわけなのです。

Human68kにもこのような機能が装備されています。マニュアルには明記されていませんが，PRNDRV?.Xを登録する際，

DEVICE=PRNDRV. X#/B64

のように指定することで，並行印字が可能になります (この場合バッファサイズは64Kバイト)。

一見便利なようなスプーラ機能ですが，ちょっとした落し穴があります。たとえば，プリントアウトを始めたはいいが，プリンタ用紙のトップフィード位置がずれていることに気がついたり，誤ってコピーキーを押してしまったりした場合，プリントアウトの中断ができないわけです。普通ならプリンタの電源を落とすなりすれば印字を止めることができますが，スプーラが設定されていると，プリンタが使用可能になった途端にバッファに溜まっているデータを出力しようとし始めます。

64Kバイトのバッファなら漢字32000字分ですから (文字なら最低でもプリンタ用紙4枚分)，打ち終わるのを待つのも大変です。ふつうはバッファをクリアする機能があるものなのですが，どうもそれが見当たりません (もしかするとあるのかもしれませんが)。そこで，任意の時点でプリンタバッファをクリアしてやろうというのが今回のプログラムです。

バッファクリアの実際

プリンタバッファはプリンタドライバ組み込み時にメモリ上に確保されます。これをクリアするにはドライバの組み込まれた位置を探し出し，バッファ部分の内容を直接書き換える……といった危険な技はとりあえず必要ありません。

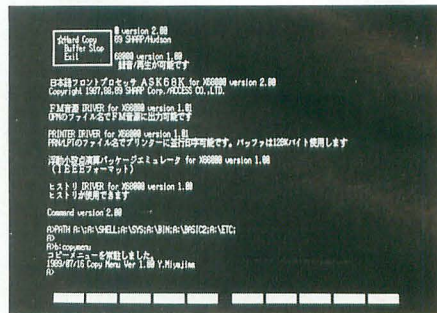
プリンタバッファを組み込んだ際のOSのデバイス状況を調べてみると，システム予約ファイルとして“SPL”というものが拡張されているのがわかります。これがプリンタスプーラ用のデバイスなのです。プリンタへ送られるデータが一時的にここに蓄えられるのでしょう。

このファイルを開いてなにか書き込んでやれば，プリンタバッファをクリアすることができるのです。

つまり，プリントアウト中に，

A>COPY CON SPL
a^Z

とか，

A>TYPE CONFIG. SYS > SPL

としてやればプリンタ出力は止まります。

コマンドモードならばこのようになんとかならないこともないのですが，まさか，BASICから，

FN=FOPEN("SPL", "W"):
FPUTC(0, FN):FCLOSE(FN)

などと打ち込む人はいないでしょうし，任意のファイルが開けないアプリケーションでは手の施しようがありません。

そこで，どこからでも簡単に扱えるようにキー割り込みを使ってこのような機能を呼び出せるようにしました。使用するキーはCOPYキーです。このキーは普段使用されることが少なく，むしろ誤って押して「プリンタがつながっていません」エラーを出されることが多いのではないでしょうか。では，従来持っていたハードコピーの機能はというと，これはコピーキーを押すことで，

Hard Copy
Buffer Clear
Exit

の各機能をメニューとして表示/選択するようにしています。なお，シフト，コントロール，OPT. 1/2キーを同時に押している場合は，それぞれの処理を行います。

単にバッファをクリアするだけならもっと簡単な処理でもいいのですが，

1) コピーキーを押す
2) メニューが開く
3) 機能を選択する
4) 本来の処理を再開する

という一連の処理は，見る人が見ればとて

もおいしい機能を予感させてくれるはずです。ここでは深く追求しませんが，今後なんらかの展開があるかもしれません。

さて，プリンタバッファが設定されていないのにこの機能を使うとどうなるでしょうか。さすがに，新たにSPLというファイルができることはありませんが，仮にSPLというファイルがすでに存在していたとすると，その内容を消去することになってしまいます。まあ，英文字3字のファイル名はシステムによって使用される可能性が高いのでユーザーは使うべきでないとマニュアルにも明記してありますが，一応気をつけてください。

プログラム解説

プログラムはリスト1，2です。マシン語入力ツールを持っている人はリスト1を打ち込み，928バイトにサイズを整えてください。そうでない人はソースを打ち込み，アセンブル/リンクします。このとき，山のようにWARNINGメッセージが出ますが気にしないでください。

コマンドラインから実行すると（バッチファイルでもいいけど），メニューが組み込まれ，COPYキーを押すたびにウィンドウが開きます。これでバッファクリアを選択するとプリンタバッファは止まるようになります。なお，この機能はプリンタを異常終了させますので使ったあとは必ずプリンタをリセットしてください。

処理の手順は次のようになっています。

●常駐手続き

TRAP #12(COPYキー) のベクタを変更

↓

必要なメモリを確保する

↓

常駐終了する

●コピーキーを押したとき

ウィンドウに隠れる部分を待避する

↓

ウィンドウを開く

●スプーラを止めるとき

スプーラをオープンする

↓

スプーラに値を出力する

↓

スプーラをクローズする

↓

ウィンドウを閉じる

実は常駐解除をできるようにしたかったのですが，TRAP #12 のベクタを元の値に戻してもシェルに戻るときにOSがTRAP #12のベクタを管理していて，ベクタを勝手に（勝手にといっちゃ悪いけど）戻してしまうのです。だから解除してからコピーキーを押すと，ヘタをすると暴走してしまいます。ということで，常駐解除はナシ。

また，一部のアプリケーションでは COPY キーのベクタなどを自分で管理しているので，この機能が使えない場合があります。いちばん困るのが，標準ワープロ。

でも，ワープロの場合は24ドットビットイメージ印字なので，1行ベッタリで2，3Kバイトのデータを必要とします。バッファ64Kバイトならプリンタ用紙1/3。これくらいは我慢することにしましょう。

おしまいに

まあ，すぐにできるだろうと踏んでいたのですが，思いのほか悩まされました。「コピーキーを押すとウィンドウが出て，カーソルキーで機能を選択できるようにする」といっても，コピーキーを押したときにキーボードの割り込みが禁止されて，TRAPが行われるらしく，カーソルキーなどのスキャンができないのです。

結局，MFPを直接操作して強引に割り込み許可にしているのですが，MFPのどこをどうすればいいのかがわからなかったので，すうぱぁ困った。それに，常駐解除するときにTRAP#12のベクタを元に戻せないとわかって，これまたすうぱぁ困った。まっ，たまにゃぁ人間苦労が大事かもね。

リスト1

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 02 02 00 00 03 32  : 39
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 00 2E 00 00 00 00  : 2E
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  60 16 59 4D 00 00 00 00  : 1C
0048  00 00 00 00 00 00 00 00  : 00
0050  00 00 00 00 53 50 4C 00  : EF
0058  48 E7 FF FE 70 04 72 0E  : 20
0060  4E 4F 02 80 00 00 00 0F  : 2E
0068  66 00 00 D6 08 F9 00 04  : 41
0070  00 E8 80 13 22 79 00 00  : 16
0078  00 04 41 F9 00 E0 00 00  : 1E
-------------------------------------
SUM:  A4 8D 1D DD ED A6 C1 53  10D0

0080  61 00 01 1A 41 F9 00 E2  : 98
0088  00 00 61 00 01 10 70 23  : 05
0090  72 00 74 00 4E 4F 23 C0  : 66
0098  00 00 00 10 70 21 43 F9  : DD
00A0  00 00 01 9C 4E 4F 70 AF  : 59
00A8  4E 4F 76 01 38 3C 81 99  : A2
00B0  61 40 70 00 4E 4F E0 48  : D6
00B8  02 80 00 00 00 FF B0 3C  : 6D
00C0  00 1D 67 3E B0 3C 00 3C  : EA
00C8  67 08 B0 3C 00 3E 67 12  : 12
00D0  60 E0 B6 3C 00 01 67 DA  : 74
00D8  38 3C 81 40 61 14 53 83  : 80
00E0  60 CA B6 3C 00 03 67 CA  : 50
00E8  38 3C 81 40 61 04 52 83  : 6F
00F0  60 BA 70 23 72 02 24 03  : 48
00F8  4E 4F 70 20 32 04 4E 4F  : 00
-------------------------------------
SUM:  C9 5F 22 7C EA EE A3 D4  4A66

0100  4E 75 41 F9 00 00 01 90  : 8E
0108  53 83 E5 8B 22 70 30 00  : 08
0110  4E D1 22 79 00 00 00 04  : BE
0118  41 F9 00 E0 00 00 61 00  : 7B
0120  00 96 41 F9 00 E2 00 00  : B2
0128  61 00 00 8C 20 39 00 00  : 46
0130  00 10 34 00 48 40 32 00  : FE
0138  70 23 4E 4F 70 AE 4E 4F  : EB
0140  4C DF 7F FF 2C 79 00 00  : 4E
0148  00 0C 4E D6 3F 3C 00 01  : AC
0150  48 79 00 00 00 14 FF 3D  : 11
0158  5C 8F 24 00 3F 02 3F 3C  : CB
0160  00 1A FF 1D 58 8F 3F 02  : 5E
```

```
0168  FF 3E 54 8F 22 79 00 00  : BB
0170  00 04 41 F9 00 E0 00 00  : 1E
0178  61 3C 41 F9 00 E2 00 00  : B9
-------------------------------------
SUM:  51 16 D1 24 1E 0E 8F 5F  6006

0180  61 34 20 39 00 00 00 10  : FE
0188  34 00 48 40 32 00 70 23  : 81
0190  4E 4F 70 AE 4E 4F 4C DF  : 83
0198  7F FF 4E 73 30 3C 00 4F  : FA
01A0  32 3C 00 08 32 D8 51 C9  : 9A
01A8  FF FC D1 FC 00 00 00 6E  : 36
01B0  51 C8 FF EE 4E 75 30 3C  : 35
01B8  00 4F 32 3C 00 08 30 D9  : CE
01C0  51 C9 FF FC D1 FC 00 00  : E2
01C8  00 6E 51 C8 FF EE 4E 75  : 37
01D0  00 00 00 D2 00 00 01 0C  : DF
01D8  00 00 01 2C 84 AC 84 AA  : 8B
01E0  84 AA 84 AA 84 AA 84 AA  : B8
01E8  84 AA 84 AA 84 AD 0D 0A  : A4
01F0  84 AB 81 40 48 61 72 64  : 6F
01F8  20 43 6F 70 79 20 81 40  : 9C
-------------------------------------
SUM:  E1 4A 71 8E 4D 4E C4 30  C780

0200  84 AB 0D 0A 84 AB 81 40  : 36
0208  42 75 66 66 65 72 20 53  : CD
0210  74 6F 70 20 84 AB 0D 0A  : B9
0218  84 AB 81 40 45 78 69 74  : 8A
0220  81 40 81 40 81 40 81 40  : 04
0228  84 AB 0D 0A 84 AF 84 AA  : A7
0230  84 AA 84 AA 84 AA 84 AA  : B8
0238  84 AA 84 AA 84 AE 0D 0A  : A5
0240  00 00 23 C8 00 00 02 90  : 7D
0248  41 E8 00 10 23 C8 00 00  : 24
0250  00 08 93 C8 2F 09 2F 08  : D2
0258  FF 4A 50 8F 70 81 93 C9  : 75
0260  4E 4F 23 C0 00 00 02 8C  : 0E
0268  70 80 72 2C 43 F9 00 00  : CA
0270  00 00 4E 4F 20 40 0C 68  : 71
0278  59 4D 00 02 67 38 23 C0  : 2A
-------------------------------------
SUM:  22 CF E3 DA 4B 4A A2 C4  C675

0280  00 00 00 0C 2F 3C 00 00  : 77
0288  0B 40 FF 48 58 8F 23 C0  : 5C
0290  00 00 00 04 70 21 43 F9  : D1
0298  00 00 02 94 4E 4F 70 81  : 24
02A0  22 79 00 00 02 8C 4E 4F  : C6
02A8  42 67 2F 3C 00 00 02 01  : 17
02B0  00 00 02 01 FF 31 70 21  : C4
```

```
02B8  43 F9 00 00 02 DF 4E 4F  : BA
02C0  70 81 22 79 00 00 02 8C  : 1A
02C8  4E 4F FF 00 00 00 00 00  : 9C
02D0  00 00 00 00 83 52 83 73  : CB
02D8  81 5B 83 81 83 6A 83 85  : D5
02E0  81 5B 82 F0 8F ED 92 93  : EF
02E8  82 B5 82 DC 82 B5 82 BD  : 0B
02F0  81 42 0D 0A 31 39 38 39  : B5
02F8  2F 30 37 2F 31 36 20 43  : 8F
-------------------------------------
SUM:  A4 C6 1E 28 C1 A4 58 4A  15BA

0300  6F 70 79 20 4D 65 6E 75  : 0D
0308  20 56 65 72 20 31 2E 30  : FC
0310  30 20 59 2E 4D 69 79 61  : 67
0318  6A 69 6D 61 0D 0A 00 83  : 3B
0320  52 83 73 81 5B 83 81 83  : AB
0328  6A 83 85 81 5B 82 CD 82  : 1F
0330  B7 82 C5 82 C9 8F ED 92  : 57
0338  93 82 B5 82 C4 82 A2 82  : B6
0340  DC 82 B7 81 42 0D 0A 31  : 20
0348  39 38 39 2F 30 37 2F 31  : A0
0350  36 20 43 6F 70 79 20 4D  : 5E
0358  65 6E 75 20 56 65 72 20  : B5
0360  31 2E 30 30 20 59 2E 4D  : B3
0368  69 79 61 6A 69 6D 61 0D  : F1
0370  0A 00 00 36 00 22 00 08  : 6A
0378  00 64 00 10 00 1A 00 18  : A6
-------------------------------------
SUM:  83 AC 4F 46 CB 43 4C EB  E7F3

0380  00 0C 00 1C 00 16 00 4C  : 8A
0388  00 04 00 04 00 6C 00 0A  : 7E
0390  00 16 00 0A 00 12 00 10  : 42
0398  00 08 00 0A 00 18 00 0A  : 34
03A0  00 00 00 00 00 00 00 00  : 00
03A8  00 00 00 00 00 00 00 00  : 00
03B0  00 00 00 00 00 00 00 00  : 00
03B8  00 00 00 00 00 00 00 00  : 00
03C0  00 00 00 00 00 00 00 00  : 00
03C8  00 00 00 00 00 00 00 00  : 00
03D0  00 00 00 00 00 00 00 00  : 00
03D8  00 00 00 00 00 00 00 00  : 00
03E0  00 00 00 00 00 00 00 00  : 00
03E8  00 00 00 00 00 00 00 00  : 00
03F0  00 00 00 00 00 00 00 00  : 00
03F8  00 00 00 00 00 00 00 00  : 00
-------------------------------------
SUM:  00 2E 00 34 00 AC 00 70  F13E
```

リスト2

```
==================  copymenu.s  ==================
  1: **********************************
  2: *                                *
  3: *       COPY MENU Ver 1.00       *
  4: *                                *
  5: *          1989/07/16            *
  6: *                                *
  7: *   Programmed By Y.Miyajima     *
  8: *                                *
  9: **********************************
 10:
 11: cpykey:
 12:     bra.s   main
 13:
 14:     dc.b    'YM'
 15: memptr:
 16:     ds.l    1
 17: proptr:
 18:     ds.l    1
 19: hcopy:
 20:     ds.l    1
 21: locate:
 22:     ds.l    1
 23: fname:
 24:     dc.b    'SPL',0
 25: main:
 26:     movem.l d0-d7/a0-a6,-(sp)
 27:
 28:     moveq.l #$04,d0             * B_BITSNS
 29:     moveq.l #$0e,d1
 30:     trap    #15
 31:     andi.l  #$0f,d0
 32:     bne     n_copy              * SHIFT,CTRLまたは,
 33:                                 * OPT.1/2が押されていたら既存のルーチンへ
 34:     bset.b  #4,$e88013          * MFP 割り込みマスクレジスタ
 35:
 36:     move.l  memptr,a1           * 待避バッファ先頭アドレス
 37:     lea     $e00000,a0          * テキストプレーン0トップ
 38:     bsr     push                * テキストプレーン0待避
 39:     lea     $e20000,a0          * テキストプレーン1トップ
 40:     bsr     push                * テキストプレーン1待避
 41:
 42:     moveq.l #$23,d0             * B_LOCATE
 43:     moveq.l #0,d1
 44:     moveq.l #0,d2
 45:     trap    #15
 46:     move.l  d0,locate
 47:
 48:     moveq.l #$21,d0             * B_PRINT
 49:     lea     window,a1
 50:     trap    #15
 51:
 52:     moveq.l #$af,d0             * OS_CUROF
 53:     trap    #15
 54:
 55:     moveq.l #1,d3               * メニューカーソルのY座標
 56:
 57: menu:
 58:     move.w  #'☆',d4
 59:     bsr     putc
 60:
 61: menu_lp:
 62:     moveq.l #$00,d0             * B_KEYINP
 63:     trap    #15
 64:     lsr     #8,d0
 65:     andi.l  #$ff,d0
 66:
 67:     cmp.b   #$1d,d0
 68:     beq     exec
 69:     cmp.b   #$3c,d0
 70:     beq     up
 71:     cmp.b   #$3e,d0
 72:     beq     down
 73:     bra     menu_lp
 74:
 75: up:
 76:     cmp.b   #1,d3
 77:     beq     menu_lp
 78:
 79:     move.w  #' ',d4
 80:     bsr     putc
 81:     subq.l  #1,d3
 82:     bra     menu
 83:
 84: down:
 85:     cmp.b   #3,d3
 86:     beq     menu_lp
 87:
 88:     move.w  #' ',d4
 89:     bsr     putc
 90:     addq.l  #1,d3
 91:     bra     menu
 92:
 93: putc:
 94:     moveq.l #$23,d0             * B_LOCATE
 95:     moveq.l #2,d1
 96:     move.l  d3,d2
 97:     trap    #15
 98:
 99:     moveq.l #$20,d0             * B_PUTC
100:     move.w  d4,d1
101:     trap    #15
102:     rts
103:
104: exec:
105:     lea     table,a0
106:     subq.l  #1,d3
107:     lsl.l   #2,d3
108:     move.l  (a0,d3),a1
109:     jmp     (a1)
110:
111: copy:
112:     move.l  memptr,a1           * 待避バッファ先頭アドレス
113:     lea     $e00000,a0          * テキストプレーン0トップ
114:     bsr     pop                 * テキストプレーン0待避
115:     lea     $e20000,a0          * テキストプレーン1トップ
116:     bsr     pop                 * テキストプレーン1待避
117:
118:     move.l  locate,d0
119:     move.w  d0,d2
120:     swap    d0
121:     move.w  d0,d1
122:     moveq.l #$23,d0             * B_LOCATE
123:     trap    #15
124:
125:     moveq.l #$ae,d0             * OS_CURON
126:     trap    #15
127: n_copy:
128:     movem.l (sp)+,d0-d7/a0-a6
129:
130:     move.l  hcopy,a6
131:     jmp     (a6)
132:
133: stop:
134:     move.w  #1,-(sp)
135:     pea     fname
136:     dc.w    $ff3d               * open
137:     addq.l  #6,sp
138:
139:     move.l  d0,d2
140:
141:     move.w  d2,-(sp)
142:     move.w  #$1a,-(sp)
143:     dc.w    $ff1d               * fputc
144:     addq.l  #4,sp
145:
146:     move.w  d2,-(sp)
147:     dc.w    $ff3e               * close
148:     addq.l  #2,sp
149:
150: exit:
151:     move.l  memptr,a1           * 待避バッファ先頭アドレス
152:     lea     $e00000,a0          * テキストプレーン0トップ
153:     bsr     pop                 * テキストプレーン0待避
154:     lea     $e20000,a0          * テキストプレーン1トップ
155:     bsr     pop                 * テキストプレーン1待避
156:
157:     move.l  locate,d0
158:     move.w  d0,d2
159:     swap    d0
160:     move.w  d0,d1
161:     moveq.l #$23,d0             * B_LOCATE
162:     trap    #15
163:
164:     moveq.l #$ae,d0             * OS_CURON
165:     trap    #15
166:
167:     movem.l (sp)+,d0-d7/a0-a6
168:     rte
169:
170: push:
171:     move.w  #79,d0
172: loop1:
173:     move.w  #8,d1
174: loop2:
175:     move.w  (a0)+,(a1)+
176:     dbra    d1,loop2
177:
178:     adda.l  #$6e,a0
179:     dbra    d0,loop1
180:     rts
181: pop:
182:     move.w  #79,d0
183: loop3:
184:     move.w  #8,d1
185: loop4:
186:     move.w  (a1)+,(a0)+
187:     dbra    d1,loop4
188:
189:     adda.l  #$6e,a0
190:     dbra    d0,loop3
191:     rts
192:
193: table:
194:     dc.l    copy
195:     dc.l    stop
196:     dc.l    exit
197: window:
198:     dc.b    '┌────────────┐',13,10
199:     dc.b    '| Hard Copy  |',13,10
200:     dc.b    '| Buffer Stop |',13,10
201:     dc.b    '| Exit       |',13,10
202:     dc.b    '└────────────┘',13,10,0
203: pend:
204:
205:     .even
206: start:
207:
208:     move.l  a0,psp
209:     lea     $10(a0),a0
210:     move.l  a0,proptr
211:     sub.l   a0,a1
212:     move.l  a1,-(sp)
213:     move.l  a0,-(sp)
214:     dc.w    $ff4a               * SETBLOCK
215:     addq.l  #8,sp
216:
217:     moveq.l #$81,d0             * B_SUPER
218:     clr.l   a1
219:     trap    #15
220:     move.l  d0,ssp
221:
222:     moveq.l #$80,d0             * B_INTVCS
223:     moveq.l #$2c,d1             * COPY KEY(TRAP 12)
224:     lea     cpykey,a1
225:     trap    #15
226:
227:     move.l  d0,a0
228:     cmp.w   #'YM',2(a0)
229:     beq     remove
230:
231:     move.l  d0,hcopy
232:
233:     move.l  #2880,-(sp)
234:     dc.w    $ff48               * MALLOC
235:     addq.l  #4,sp
236:     move.l  d0,memptr
237:
238:     moveq.l #$21,d0             * B_PRINT
239:     lea     mes1,a1
240:     trap    #15
241:
242:     moveq.l #$81,d0             * B_SUPER
243:     move.l  ssp,a1
244:     trap    #15
245:
246:     clr.w   -(sp)
247:     move.l  #pend-cpykey,-(sp)
248:     dc.w    $ff31               * KEEPPR
249:
250: remove:                         * すでに常駐していたらメッセージ
251:
252:     moveq.l #$21,d0             * B_PRINT
253:     lea     mes2,a1
254:     trap    #15
255:
256:     moveq.l #$81,d0             * B_SUPER
257:     move.l  ssp,a1
258:     trap    #15
259:
260:     dc.w    $ff00
261:
262: ssp:
263:     ds.l    1
264: psp:
265:     ds.l    1
266: mes1:
267:     dc.b    'コピーメニューを常駐しました。',13,10
268:     dc.b    '1989/07/16 Copy Menu Ver 1.00 Y.Miyajima',13,10,0
269: mes2:
270:     dc.b    'コピーメニューはすでに常駐しています。',13,10
271:     dc.b    '1989/07/16 Copy Menu Ver 1.00 Y.Miyajima',13,10,0
272:
273:     .end    start
```

PRINTER

ビデオプリンタ活用プログラム

スーパーワイドビデオコピー

Kawakami Kazuhiko
川上 和彦

CZ-6PV1をフルに生かす1.5倍拡大プリントプログラムの登場です。ビデオプリンタはビデオ信号やRGB信号しか使えないと思っていませんか？ 実はコンピュータから直接制御すれば，超多色カラープリンタとして使用できるのです。

カラーハードコピーの進歩

プリンタでもカラー対応のものが増えてきましたので，ある程度のものであれば，かなり綺麗なハードコピーがとれるようになっています。特に48ピンカラープリンタなどを使えばかなりきめ細かな出力ができます。

しかし，元になっているのはせいぜい3～4色のインクリボンですからどうしても階調が浅く，タイリングを行うため解像度も著しく低下します。遠目に眺める分には十分ですが，写真のようなクオリティを持ったフルカラーの画像がほしい場合にはちょっともの足りません。せっかく高解像度65536色を駆使して作成した画像ですから画面と同等なクオリティの出力を得たいと思う人も多いはずです。また，そういったことができなければ，65536色も宝の持ち腐れになりかねません。

ビデオプリンタの問題点

そんな声に応えてくれるのがカラービデオプリンタCZ-6PV1です。昇華性染料熱転写方式による画像はそれまでのハードコピーの常識を覆すものでした。ややハードコピーの単価が上がりますが，それに見合った良質のハードコピーがとれます。これを使えばとりあえず，画面と同等のハードコピーをとることができます。

ただ，このビデオプリンタにはランニングコストが高いという以外にも欠点があります。それはプリントアウトされるサイズが一定で，かつ小さいということです。専用の転写用紙はせいぜいハガキ大の大きさですし，しかもその全面に画像を出力できるわけではありません。通常，60×80mm，拡大モードを使用しても80×80mmが目一杯なのです。

CZ-6PV1

インクシートの消費量はプリントアウトの大きさによらず一定（1枚の出力に各色を1枚ずつ使う）ですから，これではちょっともったいない。なんとかならないものでしょうか。

文字領域の活用

ところで，このビデオプリンタには文字領域という変わったものが設定されています。これは，コンピュータと接続した場合，映像信号以外に直接コンピュータからの信号で印字を制御するエリアのことを指します（図1）。

もともと，このビデオプリンタはあらゆる用途での使用を想定されており，

1) ビデオ入力端子
2) デジタル/アナログRGB端子
3) パラレルインタフェイス（セントロニクス？）

の各入力端子を持っています。通常使用されるのは，ビデオ信号かアナログRGB信号なのですが，文字領域を使用するにはコンピュータとビデオプリンタをプリンタケーブルで接続し，画像信号を介さずにパラレル通信によって直接制御を行わなければなりません。ビデオプリンタに付属のX1用ツールではプリントアウトした映像にコメントを入れることができるようになっていました。

しかし，文字領域といっても，文字しか打ち出せないわけではありません。コンピュータで直接制御できるのですから，この部分にも画像を出力することもできるはずです。この部分まで利用すれば，従来の1.5倍の大きさでの出力ができます（ただし，印字範囲は480×480ドット。縁は除く）。

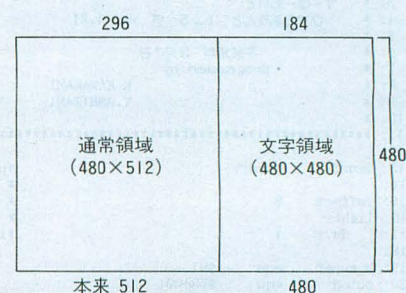
問題は画像領域に比べて文字領域は紙送りピッチが違うということです。ですから，そのまま画像を出力すると縦につぶれてしまいますし，インクの密度が違うと色の濃さがくっきりと変わってしまいます。そこで，図2のようなフォーマットで，しかも色の濃度を，私たちの目でわかる範囲で置き換えるようにしたのが今回のプログラムです。

カラーページを見てもらえばわかるように，調整次第でかなり自然に絵をつなぐことができました。これなら十分実用に耐えそうです。ただ，プリントアウトが一段と遅くなってしまいましたが。

プログラムについて

リスト2は画像領域と文字領域をすべて画像用に使用するためのプログラム（関数）です。スクリーンエディタで行番号以外の部分を打ち込んでください。

このプログラムは，Cの関数のかたちを

図1

とっています。自作のアプリケーションから呼び出す場合などに，指定すべきパラメータは，

prdrv(int light)

となっています。

よって，この関数を使ってCのプログラムを書き，それを，

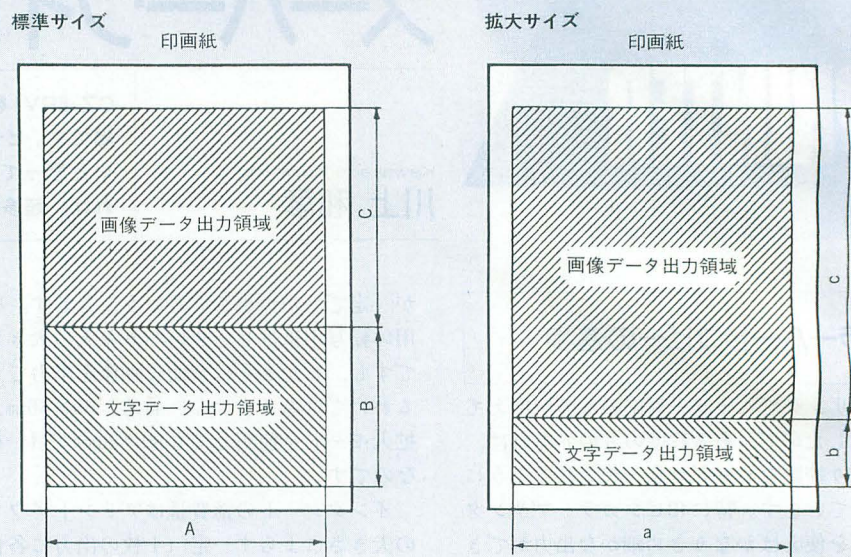
CC PROGRAM.C PRDRV.S

のようにコンパイルすれば，このプログラムで提供される機能が使用可能になります(PROGRAM.Cの部分はどんな名前でもよい)。

サンプルとして，vprint.c なるプログラムをつけておきます(リスト1)。前述のようにコンパイルしたあと，ビデオプリンタを設定し（入力はコンピュータモードで拡大モードは指定しない)，*.GL3形式の画像データを読み込ませてやります。また，参考までに，lightの値は10以下にするのが望ましいと思います（あまり大きくすると真っ白になります)。

最後に

なお，このプログラムは東京都日野市にお住まいのデザイナー芦谷氏のビデオプリンタを使わせてもらってできたものです。氏とご家族の皆さんにはいろいろ気を遣っていただいて感謝の意を表す言葉も浮かばぬほどです。本当にありがとうございました。

図2

横画素数×縦画素数

	標準サイズ		拡大サイズ	
	画像領域(A×C)	文字領域(A×B)	画像領域(a×c)	文字領域(a×b)
ビデオ入力	480×485	480×480	480×485	480×192
RGB入力	480×400	480×480	480×400	480×192
コンピュータパラレル入力	480×512	480×480	480×512	480×192

リスト1

```
=================== VPRINT.c ===================
 1: #include    "basic0.h"
 2: #include    "graph.h"
 3: #include    "image.h"
 4: static      int     light;
 5: static      unsigned char   name[32+1];
 6: static      unsigned char   a;
 7:
 8: void
 9: main()
10: {
11:     screen(1,3,1,1);
12:     printf("*** Video Print Tool ***¥n¥n");
13:     b_input("filename?(*.gl3)? ",sizeof(name),name,-1);
14:     vpage(1);
15:     img_load(name,'NASI','NASI','NASI');
16:     vpage(0);
17:     light=-1;
18:     while( light>63 | light<0 )
19:             {
20:                     b_input("どれだけあかるくしますか?(0-63):? ",0x204,&light,
-1);
21:             };
22:     vpage(1);
23:     prdrv(light);
24:     vpage(0);
25: }
```

リスト2

```
=================== prdrv.s ===================
 1: *************************************************
 2: *                                               *
 3: *   すーぱーわいど                              *
 4: *     びでおぷりんと  1.5 倍 ver 1.01           *
 5: *                                               *
 6: *           平成元年  3月7日                    *
 7: *         programmed by                         *
 8: *                          K.KAWAKAMI           *
 9: *                          T.ASHITANI           *
10: *                                               *
11: *************************************************
12:
13: .xdef       _prdrv                          *prd(int)
14:                                             * printout 480x480 picsels
15: .offset     8                               *
16: light:                                      *
17:    ds.l     1                               *int    light
18:
19: b_super     equ     $81
20: outad       equ     $e8c001
21: strob       equ     $e8c003
22: _exit       equ     $ff00
23:
24: .text
25:
26: _prdrv:
27:     link    a6,#0
28:     movem.l d3-d7/a3-a6,-(sp)
29:     bsr     _super_on
30:     bsr     print_start
31:     bsr     strob_on
32:
33:     lea     b_in,a1
34:     bsr     sc_out
35:
36:     lea     g_in,a1
37:     bsr     sc_out
38:
39:     lea     r_in,a1
40:     bsr     sc_out
41:
42:     bsr     _super_off
43:     movem.l (sp)+,d3-d7/a3-a6
```

```
 44:    unlk    a6
 45:    rts                             *return to C program
 46:
 47: ***********************************
 48: * sub routines                    *
 49: ***********************************
 50:
 51: sc_out:
 52:    move.l  #$c7fc00+2*16,a0        *Address of Screen (under left)
 53: _normalarea:
 54:    move.w  #0,d3                   *Normal_area's format is
 55:    move.w  #8-1,d0                 *((12221222122)x3+(1222))x8
 56: lp0:       move.w  #3-1,d1         *Cf1 The number is count of the same
lines.
 57: lp1:       move.w  #2-1,d2         *It makes 512lines of same lines.
 58: lp2:                               *Cf2 d3=0 <-> N.Area
 59:    bsr     line1
 60:    bsr     line2
 61:    bsr     line2
 62:    bsr     line2
 63:    dbra    d2,lp2
 64:    bsr     line1
 65:    bsr     line2
 66:    bsr     line2
 67:    dbra    d1,lp1
 68:    bsr     line1
 69:    bsr     line2
 70:    bsr     line2
 71:    bsr     line2
 72:    dbra    d0,lp0
 73:
 74: _wordsarea:                        *Words_area's format is
 75:    move.w  #1,d3                   *((23233)x4+(233))x8.
 76:    move.w  #8-1,d0                 *It makes 480 lines of 184 lines.
 77: lp3:       move.w  #4-1,d1         *Cf d3=1 <-> W.Area
 78: lp4:       move.w  #2-1,d2
 79: lp5:
 80:    bsr     line2
 81:    bsr     line3
 82:    dbra    d2,lp5
 83:    bsr     line3
 84:    dbra    d1,lp4
 85:    bsr     line2
 86:    bsr     line3
 87:    bsr     line3
 88:    dbra    d0,lp3
 89:    rts
 90:
 91: line1:                             *Draw one line
 92:    bsr     line_out
 93:    addq.l  #2,a0
 94:    rts
 95: line2:                             *Draw two same lines
 96:    bsr     line_out
 97:    bsr     line_out
 98:    addq.l  #2,a0
 99:    rts
100: line3:                             *Draw tree same lines
101:    bsr     line_out
102:    bsr     line_out
103:    bsr     line_out
104:    addq.l  #2,a0
105:    rts
106:
107: line_out:                          *Output only one line
108:    movem.l d0-d1/a0/a5,-(sp)
109:    move.l  #512*2,a5
110:    move.l  #512-1,d1
111: l_loop:
112:    jsr     (a1)
113:    bsr     io_out
114:    dbra    d1,l_loop
115:    movem.l (sp)+,d0-d1/a0/a5
116:    rts
117:
118: b_in:                              *Loading data of blue plane
119:    move.w  (a0),d0
120:    suba.l  a5,a0
121:    bsr     adjust
122:    bclr    #6,d0
123:    bset    #7,d0
124:    rts
125: g_in:
126:    move.w  (a0),d0
127:    suba.l  a5,a0
128:    ror.l   #1,d0
129:    lsr.w   #8,d0
130:    lsr.w   #2,d0
131:    ror.l   #1,d0
132:    bsr     adjust
133:    bset    #6,d0
134:    bclr    #7,d0
135:    rts
136: r_in:
137:    move.w  (a0),d0
138:    suba.l  a5,a0
139:    ror.l   #1,d0
140:    lsr.w   #5,d0
141:    rol.l   #1,d0
142:    bsr     adjust
143:    bclr    #6,d0
144:    bclr    #7,d0
145:    rts
146:
147: adjust:                            * Shift light and adjustment for Words_area
148:    bclr    #7,d0
149:    bclr    #6,d0
150:    bsr     shift
151:    bclr    #7,d0
152:    bclr    #6,d0
153:    tst     d3                      * Check (W_area->adjust,Normal_area->DO noth
ing!)
154:    beq     ad_end
155:    cmp.b   #63,d0
156:    beq     ad_end
157:    cmp.b   #51,d0
158:    beq     ad1
159:    cmp.b   #42,d0
160:    beq     ad2
161:    cmp.b   #35,d0
162:    beq     ad3
163:    cmp.b   #28,d0
164:    beq     ad4
165:    cmp.b   #21,d0
166:    beq     ad5
167:    cmp.b   #14,d0
168:    beq     ad6
169:    cmp.b   #7,d0
170:    beq     ad7
171:    addq.b  #8,d0
172:    bra     ck_flow
173: ad1:
174:    addq.b  #1,d0
175:    bra     ck_flow
176: ad2:
177:    addq.b  #2,d0
178:    bra     ck_flow
179: ad3:
180:    addq.b  #3,d0
181:    bra     ck_flow
182: ad4:
183:    addq.b  #4,d0
184:    bra     ck_flow
185: ad5:
186:    addq.b  #5,d0
187:    bra     ck_flow
188: ad6:
189:    addq.b  #6,d0
190:    bra     ck_flow
191: ad7:
192:    addq.b  #7,d0
193:
194: ck_flow:
195:    btst    #6,d0
196:    bne     flow
197: ad_end:
198:    rts
199:
200: flow:
201:    move.b  #$ff,d0
202:    rts
203:
204: shift:
205:    move.l  light(a6),d4
206:    add.b   d4,d0
207:    cmp.b   #63,d0
208:    bhi     flow
209:    rts
210:
211: io_out:                            *Output for V.P.
212:    move.b  d0,outad
213:    bsr     busy_ch
214:    rts
215:
216: _super_on:                         *Supervisermode on
217:    moveq.l #b_super,d0
218:    movea.l #0,a1
219:    trap    #15
220:    move.l  d0,ssp_save
221:    rts
222:
223: _super_off:
224:    moveq.l #b_super,d0
225:    move.l  ssp_save,a1
226:    trap    #15
227:    rts
228:
229: print_start:
230:    move.l  d0,-(sp)
231:    move.b  #%11000000,d0
232:    move.b  d0,outad
233:    move.l  (sp)+,d0
234:    rts
235:
236: busy_ch:                           *busy check
237:    btst.b  #5,$e9c001
238:    bne     strob_on
239:    bra     busy_ch
240:
241: strob_on:
242:    clr.b   strob
243: strobe_off:
244:    move.b  #$01,strob
245:    rts
246:
247: ****************************
248: *data area                 *
249: ****************************
250: .even
251: ssp_save:
252:    dc.l    0
253:
```

PRINTER

16ピンプリンタで24ドット印字を

24ピンプリンタエミュレータ

Yuasa Natsuki
湯浅 夏樹

16ピンプリンタを24ピン漢字プリンタのように使えたら……という声に応えるデバイスドライバです。ここでは対応機種としてPC-8821シリーズを対象としていますが，これを応用すればもっと多彩な機種に対応することも可能です。

X68000を買ったはよいが，24ピンドットプリンタまではお金が回らなかったという人は結構いるのではないでしょうか。僕もそのひとりです。ただ家にはPC-8822という16ピンのドットプリンタがあるので，これをなんとか生かしてみたいと考えました。そこで，プリンタドライバ作りにとりかかったのです。最初は「24ドット印字ができれば十分」と思っていたのですが，だんだん欲が出てきて，テキスト画面のハードコピーをとれるようにしたり，グラフィック画面のハードコピーをとれるようにしたりして，結構満足のいくドライバができました。

しかし，よくよく考えてみると，付属のワープロの文書もそのままでは印字できず，MUSIC PRO-68Kなどの市販ソフトでも結局プリンタは使えないままです（考えてみれば当たり前のことですが）。「こうなったらCZ系のプリンタをエミュレートしてしまおう」と思い立ち，結局このプリンタドライバができあがったのです。CZ系のプリンタをエミュレートすることにより，ワープロの文書もMUSIC PRO-68Kの楽譜も印字できるようになりましたし，おそらくほかのソフト（Z'sSTAFFなど）にも対応できると思います。

まずは使い方

リスト1がそのダンプリストです。なるべくソースを打ち込むか，デバッガでソースジェネレートしてください。PC-8821/8822を持っている人は（ほとんどいないと思いますが），これをそのまま打ち込むだけで，24ピンドットプリンタに生まれ変わります（ちょっとオーバーかな）。

このままでは，基本的にPC-8821/8822系にしか対応させていませんので，ほかの16ピンドットプリンタを持っている人はソースリストを参考にして制御コードを自分のプリンタのものに変え，ビットリバースの処理を変更すれば，やはり，24ピンドットプリンタのように使えるようになります。できるだけ細かく解説しますので，あとは各自で対応してください。

また8ドットプリンタを持っている人は，X1turboやMZ-2500などのように，半ドット改行をうまく使えば，同様の印字を行えるようになると思います（結構大変でしょうが）。

では，このプリンタドライバの使い方を説明します。

まず，CONFIG.SYSに次の1行を追加します（#以降は省略可能。またTの次は任意の2桁の数nmで，改行幅の初期設定値をnm/120インチに設定します）。

DEVICE＝NPRNDRV.SYS #/T20

#以降を省略すると改行幅は20/120（＝1/6）インチになります。

そのあと，OSを立ち上げ直すと，PRN/LPT/PPPの3つのデバイス名が登録されます。

このプリンタドライバはいくつかのモードを持っているのですが，初期状態ではPRN出力は24ドットモード，LPT出力はエミュレートモードになっています。

また，デバイスPPPに次の文字を出力するとモードが変わります。

1……PRN出力を16ドットモードにする。半角文字は16×8，全角文字は16×16で印字。

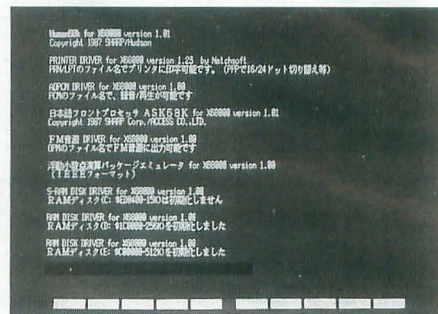
2……PRN出力を24ドットモードにする。半角文字は24×12，全角文字は24×24で印字。

E……LPT出力をエミュレートモードにする。

S……LPT出力をスモールエミュレートモードにする。

N……LPT出力をノーマルモードにする(正常に戻す)。

C……PRN/LPT出力をプリンタ出力チェックモードにする。

そのほか，PPPに0を出力すると内部のバッファなどがクリアされます。プリンタオフライン時に“A”を押した場合は，その

あとで必ず，

ECHO 0>PPP

を実行してください。そうしないと，そのあとでごみが出力されたり，コピーキーがきかなくなることがあります。

エミュレートモード時は，CZ系のプリンタのコードを認識して，できるだけ期待通りの出力がされるようにエミュレートしますが，横方向のドット数が足りないのはどうしようもないので（PC-8821/8822は960ドットしか打てないがCZ系24ドットプリンタは1440ドットも打てる），文字の出力が右端を越える場合は改行後残りを印字するようにし，ビットイメージデータの出力が右端を越える場合は越えた分を無視するようにしてあります（MUSIC PRO-68Kで楽譜をまともに印字できるようにするため）。

エミュレートの制限

このバージョンで，エミュレートを行っている機能は，以下のとおりです。

24ドットビットイメージ
n/120インチ改行設定
1/6インチ改行設定
横倍角指定（ただし漢字横倍角指定との区別はしない）
ドット単位印字位置絶対指定
ドット単位印字位置相対指定
片方向印字
両方向印字

また，次のコードは，無視しても差し支えないと思われるので，単に読み飛ばすよ

うにしてあります。

　プリンタの初期化（ESC+“c1”）

　ペーパーエンプティ無視/有効(ESC+“p0”/ESC+“p1”)

　ページ長設定（ESC+“F”）

　ページ先頭指定（ESC+“5”）

スモールエミュレートモードは，エミュレートモードの欠点をカバーするために作ったモードで，ドット数を縦横ともに2/3に圧縮することによって，横方向1440ドットの印字を可能にし，文字の大きさもCZ系24ドットプリンタで打ち出したときと同じ大きさで出力されます。

このモードで出力すれば，CZ系プリンタではだいたいこんな感じに出力されるのだなということがわかります。ただし，2/3に圧縮するので細かいところはだいぶ汚くなり，16ドットプリンタの限界を思い知らされます。通常はエミュレートモードにしておいたほうがよいでしょう（いわばレイアウト表示のプリンタ出力版のようなものです)。

なお，エミュレートモード時やノーマルモード時は，PRN出力による文字は密着していますが，このモードでは，24ドットプリンタ換算で全角時左右それぞれ6ドット(実際には2/3になるので4ドット)，半角時は右のみ6ドットのドットスペースが取られます。これは，できるだけCZ系の24ドットプリンタの出力と印字結果を似せるためです。

LPT出力を正常に戻したときは，LPTに出力したものはそのままプリンタへ出力されますが，PRN出力はこのときもエミュレートモードなので，改行幅の設定を（ESC+“Tnm”で）しても無視されてしまいます。

改行幅の設定はエミュレートモードで，ESC+“%9”+CHR$(n)で行ってください。またPPPにE,S,N,Cのどれかを出力すると改行幅は初期設定値（CONFIG.SYS内で指定した値）に戻ります。

プリンタ出力チェックモード時は，プリンタに送られたコード自体を印字します。つまり，たとえば，

　ECHO　A>PRN

とするとAの文字コードである41が印字されます。またPRNへの出力とLPTへの出力とを区別できるように，PRNへの出力はHD（高密度）パイカで，LPTへの出力はHS（高速）パイカで印字されるようにしてあります。

エミュレートモード時に未定義の制御コードがきた場合は，そのときだけ一時的にプリンタ出力チェックモードになり，その制御コードが印字されるので，プリンタドライバにその制御コードを登録し，エミュレータを成長させていくことができます（このため，このプリンタドライバは，ソースリストを入力しておかないと，この拡張ができなくなってしまうので注意してください)。

COPYキーについて

COPYキーについては機能変更があります。まず，COPYキーだけを押しても無視するようにしました。SHIFT+COPYでテキスト画面（T0プレーンのみ）のハードコピーを行います（ブレイクキーで中断できます)。

CTRL+COPYでグラフィック画面（実画面）のハードコピーを行います（ブレイクキーで中断できます)。グラフィックパレットアドレスが0以外のときはすべてドットを打ちます。

OPT.1+COPYで改頁を行います。OPT.2+COPYで改行を行います。XF1+COPYで逆方向改行します。XF2+COPYで正方向改行します。

ハードコピー時にまたハードコピーを行おうとしても無視するようにしてありますが，このためCOPYキーの処理中にプリンタオフラインなどでOSのエラー処理に制御が移り“A”で中止してしまうと，COPYキーがきかなくなってしまいます。このときは，

　ECHO　0>PPP

を実行してください。

なお，元のCOPYキーには拡大ハードコピーや色によって濃淡をつけたハードコピーをとる機能がありましたが，ドット数の関係（960ドットまでしかない）でこれらは不可能なので，これらの機能はつけてありません。

プリンタ変更時の注意

ソースリストを読めばだいたいわかると思いますが（注釈もかなり入れておきましたし），PC-8821/8822の制御コードについて，少し補足説明をしておきます。

PC-8821/8822では，16ドットのビットイメージ出力をさせるときは，ESC+“I$n_3n_2n_1n_0$”に続いて，$n_3n_2n_1n_0$×2バイトのデータを送ります（$n_3n_2n_1n_0$は4桁の10進数でドット列数)。

このときのデータは，上がLSB下がMSBで，ひとつのドット列につき，上8ドット分と下8ドット分の2バイトのデータが必要です。たとえば，

　ESC+“I0001”+CHR$(1)+CHR$(8)

と出力すると，下図のように印字されます（○はドットが打たれない部分，●はドットが打たれる部分)。

LSB ●
　　 ○
　　 ○
　　 ○
　　 ○
　　 ○
　　 ○
MSB ○
LSB ○

図1　印字サンプル(WP.X)

NPRNDRV.SYSの出力（原寸）

Ｘ６８０００を買ったは良いが、２４ドットプ
言う人は、結構いるのではないでしょうか。僕も
２２という１６ドットプリンターがあるので、こ
ました。そして今年のお正月にかなりまとまった
バー作りにとりかかりました。最初は「２４ドッ
のですが、段々欲が出てきて、テキスト画面のハ
ラフィック画面のハードコピーをとれるようにし

CZ-8PK5による出力（参考）

Ｘ６８０００を買ったは良いが、２４ドットプリンタ
言う人は、結構いるのではないでしょうか。僕もその一
２２という１６ドットプリンターがあるので、これを何
ました。そして今年のお正月にかなりまとまった時間が
バー作りにとりかかりました。最初は「２４ドット印字

○
○
●
○
○
○
MSB ○

このビット並びが逆のプリンタに移植するときは（上がMSBで下がLSBのとき）bitrevサブルーチンを，

```
bitrev: move.b d1,d0
        rts
```

に変更し，ank24，kan2，kan24各サブルーチンの，

```
roxr.b #1,d1
```

を，

```
addx.b d1,d1
```

に変更し，ank，hctext，hcgrap各サブルーチンの，

```
lsr.b #1,d1
```

を，

```
add.b d1,d1
```

に，

```
or.b #$80,d1
```

を，

```
or.b #1,d1
```

に変更します（それぞれの場所はわかりますね）。

また，lfwsetサブルーチンは，改行幅をd1/120インチに設定する制御コードをlfstcdからのメモリに代入するルーチンで，dotoutサブルーチンは，ドット列数d3の16ドットビットイメージ出力の制御コードを出力するルーチンなので，各自のプリンタに合わせて処理を変更してください。もちろん，katacd以下の制御コードを，各自のプリンタに合わせて変更するのはいうまでもありません。

これらの修正を行えば，ほかの16ドットプリンタでも，CZ系の24ドットプリンタのエミュレートが行えるようになるので，ほかの16ドットプリンタを持っている人は，ぜひ挑戦してみてください。

そのほか，移植の際に説明が必要と思われるサブルーチンについて少し説明しておきます。

__B__LPT

d1.bの内容をプリンタに出力する。

crexc

それまでに蓄えられたビットイメージデータをモードに応じて出力する。16ドットモードでビットイメージデータを含まないなら，crexc2。24ドットモードや16ドットモードでビットイメージデータを含むなら，cr24。スモールエミュレートモードならcrsm1を呼んでそれぞれの処理を行う。

crexc 2

buffに入っている内容を16ドットビットイメージで出力する。

cr24

buff2に入っている内容を16ドットビットイメージの上8ドット分のデータとして出力し，8ドット分改行したあとに，buffに入っている内容を16ドットビットイメージで出力する。

crsml

buff2とbuffに入っているデータを1ビットおきにorをとって，縦/横方向ともに2/3に圧縮してからcrexc2で出力させる。

tabout

その行の文字数が，8の倍数になるまで，スペースを出力（実際には，バッファに蓄えるだけ。crコードがくるまでは，すべてのデータはバッファに蓄えられるだけで出力はされない）。

bim???

エミュレートモード時に，ビットイメージデータをバッファに蓄えるためのサブルーチン群。

bitrev

d1.bのビット列をリバースしてd0.bに代入する。

rps???

ドット単位印字位置相対指定のための処理。

aps???

ドット単位印字位置絶対指定のための処理（やっていることはrps???と同じ）。

chkprn

d1.bの内容がわかるように出力する。たとえばd1.b=$58('X')のときは，$35，$38('58')のように16進コードに変換したものを出力。

bimchk

次の文字を印字できるかチェックし，行をはみ出すようなら改行する（実処理はbimcsbが行う）。

dot6sp

6ドット分のスペースをあける。

baiexc

いまバッファに入った文字を倍角にする。

hctext

テキストVRAMをハードコピー。

hcgrap

グラフィックVRAMをハードコピー。

lpout

a1が指すアドレスに入っている内容が0になるまでプリンタに出力し続ける。

lpoutx

a1が指すアドレスに入っている内容をd2.w文字分プリンタに出力。

lfwset

lfstcdにプリンタの改行幅をd1.b/120インチに設定するコードを代入する。

dotout

ドット列数d3.wの16ドットビットイメージを出力するためのコードをプリンタに出力する。

Profile
◇湯浅さんは東京都にお住まいの22歳，現在大学4年生です。専攻は情報工学でパソコン歴は約7年。中学1年でBASICを覚え，PC-8801を経て，MZ-2500，X68000ユーザーです。

リスト1　NPRNDRV.SYSダンプリスト

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 00 00 00 00 22 44  : 66
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 02 70 00 00 00 00  : 72
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  00 00 00 4E 80 00 00 00  : CE
0048  00 EA 00 00 00 F2 50 52  : 7E
0050  4E 20 20 20 20 20 00 00  : EE
0058  00 00 00 00 21 00 00 00  : 21
0060  01 22 00 00 01 22 00 00  : 46
0068  01 22 00 00 01 22 00 00  : 46
0070  01 22 00 00 01 22 00 00  : 46
0078  01 22 00 00 01 60 00 00  : 84
-----------------------------------
SUM:  9A E7 22 DE C5 D8 72 96  449E
```

```
0080  01 60 00 00 03 12 00 00  : 76
0088  01 28 00 00 01 22 00 00  : 4C
0090  00 9C 80 00 00 00 01 2C  : 49
0098  00 00 01 34 4C 50 54 20  : 45
00A0  20 20 20 20 00 00 00 00  : 80
00A8  00 00 21 00 00 00 01 22  : 44
00B0  00 00 01 22 00 00 01 22  : 46
00B8  00 00 01 22 00 00 01 22  : 46
00C0  00 00 01 22 00 00 01 22  : 46
00C8  00 00 01 80 00 00 01 80  : 02
00D0  00 00 03 12 00 00 01 28  : 3E
00D8  00 00 01 22 FF FF FF FF  : 1F
00E0  80 00 00 00 01 46 00 00  : C7
00E8  01 4E 50 50 50 20 20 20  : 9F
00F0  20 20 00 00 00 00 00 00  : 40
00F8  21 00 00 00 01 22 00 00  : 44
-----------------------------------
SUM:  E4 B2 1A BE A1 0B 7A 9B  8D05
```

```
0100  01 22 00 00 01 22 00 00  : 46
0108  01 22 00 00 01 22 00 00  : 46
0110  01 22 00 00 01 22 00 00  : 46
0118  01 9C 00 00 01 9C 00 00  : 3A
0120  01 28 00 00 01 28 00 00  : 52
0128  01 22 23 CD 00 00 00 16  : 29
0130  4E 75 48 E7 80 0C 2A 79  : 21
0138  00 00 00 16 49 F9 00 00  : 58
0140  00 1A 70 00 10 2D 00 02  : C9
0148  D0 40 D0 40 28 74 00 00  : BC
0150  4E 94 1B 40 00 03 E0 48  : 68
0158  1B 40 00 04 4C DF 30 01  : BB
0160  4E 75 30 3C 50 03 4E 75  : 45
0168  42 40 4E 75 23 CD 00 00  : 35
0170  00 64 4E 75 48 E7 80 0C  : E2
0178  2A 79 00 00 00 64 49 F9  : 49
-----------------------------------
SUM:  47 81 92 74 0D CD 51 54  8FF6
```

```
0180  00 00 00 68 60 BC 23 CD  : 74
0188  00 00 00 B2 4E 75 48 E7  : A4
0190  80 0C 2A 79 00 00 00 B2  : E1
0198  49 F9 00 00 00 B6 60 A2  : FA
01A0  48 E7 50 40 26 2D 00 12  : 24
01A8  67 10 22 6D 00 0E 72 00  : 86
01B0  12 19 70 3F 4E 4F 53 83  : 4D
01B8  66 F4 4C DF 02 0A 60 A8  : 99
01C0  48 E7 50 40 26 2D 00 12  : 24
01C8  67 F0 22 6D 00 0E 72 00  : 66
01D0  12 19 70 3E 4E 4F 53 83  : 4C
01D8  66 F4 60 DE 48 E7 50 40  : 57
01E0  26 2D 00 12 67 D4 22 6D  : 2F
01E8  00 0E 10 11 B0 3C 00 30  : 4B
01F0  67 34 B0 3C 00 31 67 6A  : 89
01F8  B0 3C 00 32 67 00 00 92  : 17
---------------------------------------
SUM:  54 98 5A B8 5E 2D 8E B3  A43C

0200  08 80 00 05 B0 3C 00 45  : BE
0208  67 00 00 AC B0 3C 00 53  : 52
0210  67 00 00 C6 B0 3C 00 4E  : 67
0218  67 00 00 E2 B0 3C 00 43  : 78
0220  67 00 00 F2 60 94 42 39  : C8
0228  00 00 20 C7 42 79 00 00  : A2
0230  0E F2 23 FC 00 00 0E F4  : 21
0238  00 00 20 A6 23 FC 00 00  : E5
0240  1A 36 00 00 20 AA 42 79  : D5
0248  00 00 20 AE 42 39 00 00  : 49
0250  20 C4 42 39 00 00 20 C3  : 42
0258  42 39 00 00 20 C5 60 00  : C0
0260  FF 5A 4A 39 00 00 20 9C  : 98
0268  67 00 FF 50 42 39 00 00  : 31
0270  20 9C 13 F9 00 00 20 9E  : 86
0278  00 00 20 9D 61 00 0C 08  : 32
---------------------------------------
SUM:  B4 9B 41 BA AA DA 5E D4  CEDC

0280  42 39 00 00 20 C7 42 39  : DD
0288  00 00 0E F3 60 00 FF 2C  : 8C
0290  4A 39 00 00 20 9C 66 00  : A5
0298  FF 22 50 F9 00 00 20 9C  : 26
02A0  13 F9 00 00 20 9F 00 00  : CB
02A8  20 9D 43 F9 00 00 20 CF  : E8
02B0  61 00 0C 12 60 C6 50 F9  : EE
02B8  00 00 20 C2 33 F9 00 00  : 0E
02C0  20 A2 00 00 20 A0 42 39  : FD
02C8  00 00 20 C7 43 F9 00 00  : 23
02D0  20 CF 61 00 0B F0 60 42  : ED
02D8  13 FC 00 01 00 00 20 C2  : F2
02E0  33 F9 00 00 20 A4 00 00  : F0
02E8  20 A0 50 F9 00 00 20 9C  : C5
02F0  13 F9 00 00 20 9F 00 00  : CB
02F8  20 9D 60 CA 42 39 00 00  : 62
---------------------------------------
SUM:  F8 C6 FE 44 43 C6 19 A2  5736

0300  20 C7 42 39 00 00 20 C2  : 44
0308  33 F9 00 00 20 A2 00 00  : EE
0310  20 A0 60 06 50 F9 00 00  : 6F
0318  20 C7 42 B9 00 00 20 B2  : B4
0320  42 79 00 00 20 B6 42 79  : 4C
0328  00 00 20 BA 42 39 00 00  : 55
0330  20 BD 42 79 00 00 20 C0  : 78
0338  42 39 00 00 20 C3 42 39  : D9
0340  00 00 20 C4 12 39 00 00  : 2F
0348  20 D2 61 00 0B 92 60 00  : 50
0350  FE 6A 70 3D 4E 4F 4A 00  : FC
0358  66 00 FE 0E 70 01 4E 75  : A6
0360  20 39 00 ED 00 3C 08 39  : C3
0368  00 05 00 E9 C0 01 66 2A  : 3F
0370  4A 39 00 00 20 C6 66 20  : EF
0378  53 80 66 EA 48 E7 7F FE  : CF
---------------------------------------
SUM:  78 C9 9B FA F5 52 2F DC  699E

0380  3E 3C 70 09 4E 4E 30 07  : C6
0388  4C DF 7F FE B0 3C 00 02  : 96
0390  66 CE 50 F9 00 00 20 C6  : 63
0398  4E 75 42 39 00 00 20 C6  : 24
03A0  13 C1 00 E8 C0 01 42 39  : F8
03A8  00 E8 C0 03 61 08 13 FC  : 23
03B0  00 01 00 E8 C0 03 4E 75  : 6F
03B8  4A 39 00 00 20 C7 66 00  : D0
03C0  05 28 4A 39 00 00 20 C1  : 91
03C8  66 08 4A 39 00 00 20 C2  : D3
03D0  67 8E 4A 79 00 00 20 B2  : 8A
03D8  66 00 03 D2 4A 39 00 00  : BE
03E0  20 B6 66 00 03 22 4A 39  : E4
03E8  00 00 20 B7 66 00 03 56  : 96
03F0  4A 39 00 00 20 BA 66 00  : C3
03F8  02 D8 4A 39 00 00 20 BB  : 38
---------------------------------------
SUM:  3F C6 F2 B9 D2 72 AC BE  CAAF

0400  66 00 04 2A 4A 39 00 00  : 17
0408  20 BD 66 00 04 94 4A 39  : 5E
0410  00 00 20 C0 66 00 04 B6  : 00
0418  4A 39 00 00 20 B4 66 00  : BD
0420  02 20 4A 39 00 00 20 B5  : 7A
0428  66 00 01 F4 B2 3C 00 20  : 69
0430  64 4A B2 3C 00 0A 67 00  : 0D
0438  0A 32 B2 3C 00 0C 67 00  : 9D
0440  0A 38 B2 3C 00 0D 67 52  : F6
0448  B2 3C 00 09 67 00 01 82  : E1
0450  B2 3C 00 1B 67 00 01 A8  : 19
0458  B2 3C 00 1C 67 00 01 A8  : 1A
0460  B2 3C 00 0E 67 00 01 A8  : 0C
0468  B2 3C 00 0F 67 00 01 A8  : 0D
0470  B2 3C 00 1A 67 00 09 F4  : 6C
0478  60 00 04 6E 4A 39 00 00  : 55
---------------------------------------
SUM:  3C 32 EF B0 3A 19 17 2C  446E

0480  20 C1 66 00 04 EC 0C 39  : 7C
0488  00 01 00 00 20 C2 67 00  : 4A
0490  04 E0 4E 75 61 04 60 00  : 6C
0498  09 D2 48 E7 70 40 0C 39  : FF
04A0  00 01 00 00 20 C2 67 00  : 4A
04A8  00 A4 4A 39 00 00 20 C4  : 0B
04B0  67 0A 4A 39 00 00 20 C2  : D6
04B8  66 00 00 88 4A 39 00 00  : 71
04C0  20 9C 66 2E 36 39 00 00  : BF
04C8  20 AE 67 1C 43 F9 00 00  : 8D
04D0  20 D3 61 00 09 F0 61 00  : AE
04D8  0A 2C 43 F9 00 00 0E F4  : 74
04E0  24 03 D4 42 61 00 09 EA  : 91
04E8  61 00 09 AC 4C DF 02 0E  : 51
04F0  4E 75 36 39 00 00 20 AE  : 00
04F8  67 2E 43 F9 00 00 20 DD  : CE
---------------------------------------
SUM:  9E 12 57 B9 8E EE 40 6F  8D07

0500  61 00 09 C2 61 00 09 FE  : 94
0508  43 F9 00 00 1A 36 34 03  : C3
0510  53 42 12 19 61 00 FE 4A  : 69
0518  42 01 61 00 FE 44 51 CA  : 01
0520  FF F2 61 00 09 40 60 A4  : 9F
0528  43 F9 00 00 20 DD 61 00  : 9A
0530  09 94 61 00 09 30 43 F9  : 73
0538  00 00 20 D3 61 00 09 86  : E3
0540  60 A6 0C 39 00 01 00 00  : 4C
0548  20 C2 66 A6 36 39 00 00  : 5D
0550  20 AE 67 94 22 79 00 00  : 64
0558  20 A6 42 51 2F 0A 43 F9  : CE
0560  00 00 0E F4 45 F9 00 00  : 40
0568  1A 36 53 43 72 07 34 11  : A4
0570  E1 5A D4 42 D1 40 B2 7C  : 90
0578  00 02 66 04 14 1A E1 5A  : D5
---------------------------------------
SUM:  3F 09 14 EF 90 DE A3 18  3E47

0580  D4 42 D1 40 D4 42 64 04  : A5
0588  80 7C 00 01 51 C9 FF E4  : FA
0590  E1 58 32 C0 51 CB FF D6  : 1C
0598  70 00 30 39 00 00 20 AE  : A7
05A0  53 40 80 FC 00 03 76 01  : 89
05A8  D6 40 43 F9 00 00 0E F4  : 54
05B0  24 49 32 DA 32 1A 82 5A  : A1
05B8  32 C1 51 C8 FF F6 24 5F  : 84
05C0  48 40 D6 43 4A 40 66 00  : 91
05C8  FF 04 53 43 60 00 FE FE  : F5
05D0  48 E7 60 60 14 39 00 00  : 3C
05D8  20 C3 42 39 00 00 20 C3  : 41
05E0  72 20 61 00 03 8C 10 39  : CB
05E8  00 00 0E F2 C0 3C 00 07  : 03
05F0  66 EE 13 C2 00 00 20 C3  : 0C
05F8  4C DF 06 06 4E 75 50 F9  : 43
---------------------------------------
SUM:  F7 7B CC AA 76 9F B0 D7  4179

0600  00 00 20 B4 4E 75 50 F9  : E0
0608  00 00 20 B5 4E 75 50 F9  : E1
0610  00 00 20 C3 4E 75 42 39  : 21
0618  00 00 20 C3 4E 75 42 39  : 21
0620  00 00 20 B5 B2 3C 00 70  : 33
0628  67 E4 B2 3C 00 71 67 E6  : F7
0630  3F 01 12 3C 00 1C 61 00  : 0B
0638  02 B0 32 1F 60 00 02 AA  : 0F
0640  42 39 00 00 20 B4 B2 3C  : 3D
0648  00 25 67 6E B2 3C 00 4A  : 32
0650  67 70 B2 3C 00 63 67 78  : 07
0658  B2 3C 00 70 67 72 B2 3C  : 25
0660  00 5C 67 74 B2 3C 00 10  : 35
0668  67 76 B2 3C 00 46 67 00  : 78
0670  00 80 B2 3C 00 36 67 00  : 0B
0678  00 80 B2 3C 00 66 67 2A  : 65
---------------------------------------
SUM:  6A 71 2C 7D 35 80 EE D8  7B35

0680  B2 3C 00 72 67 24 B2 3C  : D9
0688  00 5D 67 1E B2 3C 00 3E  : 0E
0690  67 18 B2 3C 00 35 67 10  : 19
0698  3F 01 12 3C 00 1B 61 00  : 0A
06A0  02 48 32 1F 60 00 02 42  : 3F
06A8  4E 75 3F 01 12 3C 00 1B  : 6C
06B0  61 00 FC AE 32 1F 60 00  : BC
06B8  FC A8 50 F9 00 00 20 B6  : C3
06C0  4E 75 50 F9 00 00 20 B7  : E3
06C8  50 F9 00 00 20 C4 4E 75  : F0
06D0  57 F9 00 00 20 BA 4E 75  : ED
06D8  50 F9 00 00 20 BB 4E 75  : E7
06E0  13 FC 00 04 00 00 20 BD  : F0
06E8  42 79 00 00 20 BE 4E 75  : 5C
06F0  50 F9 00 00 20 C0 4E 75  : EC
06F8  3F 01 12 3C 00 14 61 00  : 03
---------------------------------------
SUM:  2E E6 4A 08 5D D6 23 5A  D924

0700  07 DE 32 1F 4E 75 0C 39  : 3E
0708  00 02 00 00 20 B6 67 2A  : 69
0710  B2 3C 00 39 66 0A 13 FC  : A6
0718  00 02 00 00 20 B6 4E 75  : 9B
0720  42 39 00 00 20 B6 2F 01  : 81
0728  72 1B 61 00 01 BC 72 25  : 42
0730  61 00 01 B6 22 1F 60 00  : B9
0738  01 B0 42 39 00 00 20 B6  : 02
0740  60 00 07 9C 0C 39 00 02  : 4A
0748  00 00 20 B7 67 10 13 C1  : 22
0750  00 00 20 B8 13 FC 00 02  : E9
0758  00 00 20 B7 4E 75 48 E7  : C9
0760  60 00 42 39 00 00 20 B7  : B2
0768  E1 59 12 39 00 00 20 B8  : 5D
0770  E1 59 33 C1 00 00 20 B2  : 00
0778  D3 79 00 00 20 AE 33 C1  : 0E
---------------------------------------
SUM:  24 4D C4 3C 2B E4 E3 3E  BF69

0780  00 00 20 B0 42 39 00 00  : 4B
0788  20 B9 34 39 00 00 20 A0  : 06
0790  32 39 00 00 20 AE 92 42  : 0D
0798  63 0C 93 79 00 00 20 B0  : 4B
07A0  33 C2 00 00 20 AE 4C DF  : EE
07A8  00 06 4E 75 48 E7 70 40  : A8
07B0  4A 39 00 00 20 B9 66 24  : E6
07B8  4A 79 00 00 20 B0 67 10  : 0A
07C0  61 5E 22 79 00 00 20 AA  : 24
07C8  12 C0 23 C9 00 00 20 AA  : 88
07D0  52 39 00 00 20 B9 4C DF  : 8F
07D8  02 0E 4E 75 4A 79 00 00  : 96
07E0  20 B0 67 10 61 3A 22 79  : 7D
07E8  00 00 20 A6 12 C0 23 C9  : 84
07F0  00 00 20 A6 52 39 00 00  : 51
07F8  20 B9 0C 39 00 02 00 00  : 20
---------------------------------------
SUM:  83 46 7B 23 39 4C 2C 5A  6888

0800  20 B9 67 D2 42 39 00 00  : 8D
0808  20 B9 4A 79 00 00 20 B0  : 6C
0810  67 06 53 79 00 00 20 B0  : 09
0818  53 79 00 00 20 B2 60 B6  : B4
0820  74 07 E2 09 E3 10 51 CA  : 74
0828  FF FA 4E 75 0C 39 00 02  : 03
0830  00 00 20 BB 67 10 13 C1  : 26
0838  00 00 20 BC 13 FC 00 02  : ED
0840  00 00 20 BB 4E 75 48 E7  : CD
0848  40 60 42 39 00 00 20 BB  : F6
0850  E1 59 12 39 00 00 20 BC  : 61
0858  D3 79 00 00 20 AE 30 39  : 83
0860  00 00 20 AE B0 79 00 00  : F7
0868  20 A0 62 28 22 79 00 00  : E5
0870  20 A6 24 79 00 00 20 AA  : 2D
0878  60 04 42 59 42 1A 51 C9  : 75
---------------------------------------
SUM:  01 6E D0 8E 4D 6F 2D AF  BC07

0880  FF FA 23 C9 00 00 20 A6  : AB
0888  23 CA 00 00 20 AA 4C DF  : E2
0890  06 02 4E 75 93 79 00 00  : D7
0898  20 AE 61 00 FB F8 60 EE  : 70
08A0  48 E7 40 60 30 39 00 00  : 38
08A8  20 BE C0 FC 00 0A C2 7C  : E2
08B0  00 FF 92 3C 00 30 D2 40  : 0F
08B8  53 39 00 00 20 BD 67 98  : 68
08C0  33 C1 00 00 20 BE 4C DF  : FD
08C8  06 02 4E 75 0C 39 00 02  : 12
08D0  00 00 20 C0 67 0A 13 FC  : 60
08D8  00 02 00 00 20 C0 4E 75  : A5
08E0  42 39 00 00 20 C0 4E 75  : 1E
08E8  48 E7 60 40 24 01 72 1B  : 81
08F0  61 00 FA 6E 72 48 4A 39  : 06
08F8  00 00 20 C1 66 02 72 4E  : 09
---------------------------------------
SUM:  27 36 4C 7A CD 17 F0 30  0891

0900  61 00 FA 5E 22 02 E8 09  : CE
0908  61 1A 22 02 C2 3C 00 0F  : AC
0910  61 12 74 07 43 F9 00 00  : 2A
0918  20 C8 61 00 05 B4 4C DF  : 2D
0920  02 06 4E 75 B2 3C 00 0A  : C3
0928  65 02 5E 01 D2 3C 00 30  : 04
0930  60 00 FA 2E 2F 01 50 F9  : 01
0938  00 00 20 C1 E0 59 B2 3C  : 08
0940  00 80 65 16 B2 3C 00 A0  : 89
0948  65 06 B2 3C 00 E0 65 0A  : A8
0950  C2 BC 00 00 00 FF 61 00  : DE
0958  FA 60 22 17 C2 BC 00 00  : 11
```

```
0960   00 FF 61 00 FA 54 42 39   : 29
0968   00 00 20 C1 22 1F 4E 75   : E5
0970   4A 39 00 00 20 9C 67 08   : AE
0978   48 41 32 3C 00 0C 48 41   : 8C
-----------------------------------------
SUM:   BD 17 A3 32 6F AF 3B 07   4B7E

0980   4A 39 00 00 0E F3 66 00   : EA
0988   00 C0 B2 3C 00 80 65 14   : A7
0990   B2 3C 00 A0 65 06 B2 3C   : E7
0998   00 E0 65 08 13 C1 00 00   : 21
09A0   0E F3 4E 75 48 E7 38 F0   : 1B
09A8   43 F9 00 00 1F D6 70 19   : BA
09B0   4E 4F 0C 51 00 0C 67 52   : BF
09B8   61 00 01 86 24 79 00 00   : 85
09C0   20 A6 26 79 00 00 20 AA   : 2F
09C8   74 07 20 49 76 01 78 07   : DA
09D0   E2 09 05 18 67 04 82 3C   : 31
09D8   00 80 51 CC FF F4 14 C1   : 65
09E0   51 CB FF EC 42 1B 52 79   : 2F
09E8   00 00 20 AE 51 CA FF DC   : C4
09F0   23 CA 00 00 20 A6 23 CB   : A1
09F8   00 00 20 AA 52 39 00 00   : 55
-----------------------------------------
SUM:   E6 1B 4D 1A F2 39 2E 79   36E2

0A00   0E F2 4C DF 0F 1C 60 00   : B6
0A08   00 B6 61 00 01 34 24 79   : E9
0A10   00 00 20 A6 26 79 00 00   : 65
0A18   20 AA 74 0B 20 49 76 07   : 2F
0A20   E3 D8 E2 11 51 CB FF FA   : C3
0A28   16 C1 78 01 76 07 E3 D8   : 88
0A30   E2 11 51 CB FF FA 14 C1   : DD
0A38   51 CC FF F2 52 79 00 00   : D9
0A40   20 AE 51 CA FF D8 60 A8   : C8
0A48   48 E7 38 F0 E1 59 12 39   : DC
0A50   00 00 0E F3 E1 59 43 F9   : 77
0A58   00 00 1F D6 70 19 4E 4F   : 1B
0A60   30 11 B0 7C 00 08 67 00   : DC
0A68   FF 50 B0 7C 00 0C 67 9A   : 88
0A70   B0 7C 00 18 67 6A 61 00   : 76
0A78   00 C8 24 79 00 00 20 A6   : 2B
-----------------------------------------
SUM:   A1 02 25 6B 06 78 42 7C   51CC

0A80   26 79 00 00 20 AA 74 0F   : EC
0A88   20 49 76 01 78 07 E3 D8   : 1A
0A90   E2 11 51 CC FF FA 14 C1   : DE
0A98   51 CB FF F2 42 1B 52 79   : 35
0AA0   00 00 20 AE 51 CA FF E2   : CA
0AA8   23 CA 00 00 20 A6 23 CB   : A1
0AB0   00 00 20 AA 54 39 00 00   : 57
0AB8   0E F2 4C DF 0F 1C 0C 39   : 9B
0AC0   00 01 00 00 20 C2 66 04   : 4D
0AC8   61 00 00 8E 4A 39 00 00   : 72
0AD0   20 C3 67 04 61 00 01 1C   : CC
0AD8   42 39 00 00 0E F3 4E 75   : 3F
0AE0   61 5E 45 F9 00 00 20 22   : 3F
0AE8   72 17 14 D9 14 D9 14 D9   : 50
0AF0   52 8A 51 C9 FF F6 43 F9   : 27
0AF8   00 00 20 22 24 79 00 00   : DF
-----------------------------------------
SUM:   92 56 83 45 BD C1 17 90   B867

0B00   20 A6 26 79 00 00 20 AA   : 2F
0B08   74 17 20 49 76 07 20 10   : A1
0B10   D0 80 20 C0 E2 11 51 CB   : 3F
0B18   FF F6 16 C1 78 01 76 07   : C2
0B20   20 10 D0 80 20 C0 E2 11   : 53
0B28   51 CB FF F6 14 C1 51 CC   : 03
0B30   FF EE 52 79 00 00 20 AE   : 86
0B38   51 CA FF D0 60 00 FF 6A   : B3
0B40   61 48 0C 39 00 01 00 00   : EF
0B48   20 C2 66 08 0C 51 00 1C   : C9
0B50   65 02 61 04 58 89 4E 75   : 70
0B58   48 E7 40 60 22 79 00 00   : 6A
0B60   20 A6 24 79 00 00 20 AA   : 2D
0B68   72 02 42 99 42 5A 54 79   : B8
0B70   00 00 20 AE 51 C9 FF F4   : DB
0B78   23 C9 00 00 20 A6 23 CA   : 9F
-----------------------------------------
SUM:   07 2A 35 67 9D B7 3D F3   0DAF

0B80   00 00 20 AA 4C DF 06 02   : FD
0B88   4E 75 0C 39 00 01 00 00   : 09
0B90   20 C2 66 0A 5C 51 0C 51   : 5C
0B98   00 16 65 02 5C 51 30 11   : 6B
0BA0   4A 39 00 00 20 C3 67 02   : CF
0BA8   D0 40 D0 79 00 00 20 AE   : 27
0BB0   B0 79 00 00 20 A0 62 00   : 4B
0BB8   F8 DC 70 01 0C 39 00 01   : 8B
0BC0   00 00 20 C2 66 08 0C 51   : AD
0BC8   00 1C 65 0A 60 06 0C 51   : 4E
0BD0   00 10 65 02 70 02 4A 39   : 6C
0BD8   00 00 20 C3 67 02 D0 00   : 1C
0BE0   D0 39 00 00 0E F2 B0 39   : F2
0BE8   00 00 20 9D 62 00 F8 A6   : BD
0BF0   4E 75 48 E7 00 78 52 39   : F5
0BF8   00 00 0E F2 30 39 00 00   : 69
-----------------------------------------
SUM:   4E F5 B7 70 8D D3 57 08   9589

0C00   1F D6 0C 39 00 01 00 00   : 3B
0C08   20 C2 66 08 B0 7C 00 1C   : 98
0C10   65 0E 60 06 B0 7C 00 10   : 15
0C18   65 06 52 39 00 00 0E F2   : F6
0C20   22 79 00 00 20 A6 24 79   : FE
0C28   00 00 20 AA 26 49 28 4A   : AB
0C30   D2 C0 D2 C0 D4 C0 23 C9   : A4
0C38   00 00 20 A6 23 CA 00 00   : B3
0C40   20 AA 53 40 33 23 33 13   : F9
0C48   15 24 15 14 52 79 00 00   : 2D
0C50   20 AE 51 C8 FF F0 4C DF   : 01
0C58   1E 00 4E 75 4A F9 00 00   : 24
0C60   20 C5 66 00 00 E2 48 E7   : 5C
0C68   FF FC 2E 00 72 0A 70 04   : 19
0C70   4E 4F 2C 00 00 39 00 18   : 1A
0C78   00 E8 80 13 08 07 00 03   : 8D
-----------------------------------------
SUM:   DD 59 7D 34 E5 23 B4 A2   3F45

0C80   66 00 01 B6 08 07 00 02   : 2E
0C88   66 00 01 B4 08 07 00 00   : 2A
0C90   66 1C 08 07 00 01 66 00   : F8
0C98   00 B0 08 06 00 05 66 00   : 29
0CA0   01 A4 08 06 00 06 66 00   : 1F
0CA8   01 AC 60 00 00 90 61 00   : FE
0CB0   01 D6 43 F9 00 00 20 CF   : 02
0CB8   61 00 02 0A 43 F9 00 00   : A9
0CC0   20 D8 61 00 02 00 20 7C   : F7
0CC8   00 E0 00 00 36 39 00 00   : 4F
0CD0   20 86 45 F9 00 00 0E F4   : E6
0CD8   28 48 38 39 00 00 20 84   : 85
0CE0   7A 07 26 4C 7C 01 7E 07   : F5
0CE8   E2 09 0B 13 67 04 82 3C   : 32
0CF0   00 80 47 EB 00 80 51 CF   : 52
0CF8   FF F0 14 C1 51 CE FF E8   : CA
-----------------------------------------
SUM:   59 F8 29 BD BF 2F 51 BF   6D38

0D00   72 0C 70 04 4E 4F 08 00   : 97
0D08   00 01 66 30 51 CD FF D4   : 88
0D10   52 8C 51 CC FF CC 43 F9   : 02
0D18   00 00 20 E8 61 00 01 A6   : 10
0D20   34 39 00 00 20 82 43 F9   : 4B
0D28   00 00 0E F4 61 00 01 A2   : 06
0D30   61 00 01 32 41 E8 08 00   : C5
0D38   51 CB FF 98 4C DF 3F FF   : 1C
0D40   42 39 00 00 20 C5 4E 73   : 21
0D48   72 FF 70 B1 4E 4F 4A 80   : F9
0D50   6B EA 61 00 01 32 36 39   : 58
0D58   00 00 20 94 7E 00 3E 39   : A9
0D60   00 00 20 98 4B F9 00 00   : FC
0D68   20 EF 33 F9 00 00 20 8A   : E5
0D70   00 00 20 88 33 F9 00 00   : D4
0D78   20 90 00 00 20 8E 72 FF   : CF
-----------------------------------------
SUM:   09 3E B9 04 98 F7 74 FB   5BDF

0D80   70 10 4E 4F B0 3C 00 10   : 19
0D88   67 2C B0 3C 00 04 65 26   : 0E
0D90   36 39 00 00 20 96 3E 39   : 9C
0D98   00 00 20 9A 4B F9 00 00   : FE
0DA0   20 F6 33 F9 00 00 20 8C   : EE
0DA8   00 00 20 88 33 F9 00 00   : D4
0DB0   20 92 00 00 20 8E 43 F9   : 9C
0DB8   00 00 20 CF 61 00 01 06   : 57
0DC0   43 F9 00 00 20 D8 61 00   : 95
0DC8   00 FC 61 00 00 98 20 7C   : 91
0DD0   00 C0 00 00 45 F9 00 00   : FE
0DD8   0E F4 28 48 38 39 00 00   : E3
0DE0   20 8E 26 4C 7A 01 7C 07   : 1E
0DE8   E2 09 4A 53 67 04 82 3C   : B1
0DF0   00 80 D7 C7 51 CE FF F2   : 2E
0DF8   14 C1 51 CD FF EA 54 8C   : BC
-----------------------------------------
SUM:   B4 7E B2 F0 9D B5 D9 37   0E54

0E00   72 0C 70 04 4E 4F 08 00   : 97
0E08   00 01 66 00 FF 30 51 CC   : B3
0E10   FF D2 22 4D 61 00 00 AE   : 4F
0E18   34 39 00 00 20 88 43 F9   : 51
0E20   00 00 0E F4 61 00 00 AA   : 0D
0E28   61 3A E9 8F D1 C7 E8 8F   : 22
0E30   51 CB FF A2 60 00 FF 06   : 22
0E38   61 30 60 00 FF 00 61 38   : 89
0E40   60 00 FE FA 61 40 43 F9   : 35
0E48   00 00 20 E2 61 76 61 14   : 4E
0E50   60 00 FE EA 61 30 43 F9   : 15
0E58   00 00 20 E5 61 66 61 04   : 31
0E60   60 00 FE DA 2F 01 61 2E   : F7
0E68   60 04 2F 01 61 18 72 0A   : 89
0E70   61 00 F4 EE 22 1F 4E 75   : 47
0E78   2F 01 61 0A 72 0C 61 00   : 7A
-----------------------------------------
SUM:   C8 52 0C F4 07 5E AE A1   5B88

0E80   F4 E0 22 1F 4E 75 4A 79   : 9B
0E88   00 00 20 AE 67 04 60 00   : 99
0E90   F6 0A 60 00 00 02 72 0D   : E1
0E98   61 00 F4 C6 23 FC 00 00   : 3A
0EA0   0E F4 00 00 20 A6 23 FC   : E7
0EA8   00 00 1A 36 00 00 20 AA   : 1A
0EB0   42 79 00 00 20 AE 42 39   : 04
0EB8   00 00 0E F2 42 39 00 00   : 7B
0EC0   20 C4 4E 75 12 19 67 06   : 3F
0EC8   61 00 F4 96 60 F6 4E 75   : 04
0ED0   53 42 12 19 61 00 F4 8A   : 9F
0ED8   51 CA FF F8 4E 75 48 E7   : 04
0EE0   60 40 22 7C 00 00 20 D5   : 33
0EE8   74 00 14 01 84 FC 00 0A   : 13
0EF0   D4 3C 00 30 12 C2 48 42   : 9E
0EF8   D4 3C 00 30 12 82 4C DF   : FF
-----------------------------------------
SUM:   3C DF 47 B4 23 C8 46 51   01CA

0F00   02 06 4E 75 43 F9 00 00   : 07
0F08   20 FD 61 B8 72 00 32 03   : DD
0F10   82 FC 03 E8 61 0E 82 FC   : 56
0F18   00 64 61 08 82 FC 00 0A   : 55
0F20   61 00 00 02 D2 3C 00 30   : A1
0F28   61 00 F4 36 42 41 48 41   : 97
0F30   4E 75 00 00 00 00 00 00   : C3
0F38   00 00 00 00 00 00 00 00   : 00
0F40   00 00 00 00 00 00 00 00   : 00
0F48   00 00 00 00 00 00 00 00   : 00
0F50   00 00 00 00 00 00 00 00   : 00
0F58   00 00 00 00 00 00 00 00   : 00
0F60   00 00 00 00 00 00 00 00   : 00
0F68   00 00 00 00 00 00 00 00   : 00
0F70   00 00 00 00 00 00 00 00   : 00
0F78   00 00 00 00 00 00 00 00   : 00
-----------------------------------------
SUM:   B4 D8 07 55 AC 80 FC 7A   C112
```

（207F$_H$まで0で埋める）

```
2080   00 00 00 00 00 00 00 00   : 00
2088   00 00 00 00 00 00 00 00   : 00
2090   00 00 00 00 00 00 00 00   : 00
2098   00 00 00 00 00 00 00 00   : 00
20A0   00 00 00 00 00 00 00 00   : 00
20A8   00 00 00 00 00 00 00 00   : 00
20B0   00 00 00 00 00 00 00 00   : 00
20B8   00 00 00 00 00 00 00 00   : 00
20C0   00 00 06 00 00 5F 00 1F   : 84
20C8   07 80 07 80 04 00 03 BF   : D4
20D0   03 BF 01 FF 00 3F 00 1F   : 20
20D8   08 00 04 00 FF 50 78 50   : 23
20E0   03 C0 03 C0 05 A0 00 00   : 2B
20E8   0E F4 00 00 1A 36 00 00   : 52
20F0   00 00 00 00 00 00 00 00   : 00
20F8   00 00 00 00 00 00 00 00   : 00
-----------------------------------------
SUM:   23 F3 15 3F 22 C4 7B 4D   1328

2100   00 00 FF 00 00 00 00 00   : FF
2108   1B 53 30 30 30 31 00 1B   : 4A
2110   3E 00 14 1B 54 32 30 00   : 23
2118   1B 54 31 36 00 1B 54 30   : 75
2120   38 00 1B 72 00 1B 66 00   : 46
2128   1B 49 30 37 36 38 00 1B   : 54
2130   49 30 39 36 30 00 1B 49   : 7C
2138   30 35 31 32 00 1B 49 00   : 2C
2140   48 E7 40 C0 22 6D 00 12   : D0
2148   10 19 66 FC 10 19 67 38   : 53
2150   B0 3C 00 23 66 F6 10 19   : 94
2158   67 2E B0 3C 00 2F 66 6A   : 80
2160   10 19 C0 7C 00 DF B0 3C   : 30
2168   00 54 66 5E 72 00 12 19   : B5
2170   92 3C 00 30 C2 FC 00 0A   : C6
2178   D2 19 92 3C 00 30 13 C1   : BD
-----------------------------------------
SUM:   23 81 37 F3 B6 A2 00 9C   2E3B

2180   00 00 20 D2 61 00 ED 58   : 98
2188   43 F9 00 00 0C 1C 72 2C   : 02
2190   70 80 4E 4F 43 F9 00 00   : C9
2198   03 78 32 3C 01 3E 70 80   : 18
21A0   4E 4F 43 F9 00 00 08 F4   : D5
21A8   32 3C 01 3F 70 80 4E 4F   : 3B
21B0   48 79 00 00 21 9E FF 09   : 88
21B8   58 8F 2B 7C 00 00 21 00   : AF
21C0   00 0E 4C DF 03 02 42 40   : C0
21C8   4E 75 48 79 00 00 22 24   : CA
21D0   FF 09 58 8F 4C DF 03 02   : 1F
21D8   30 3C 70 0D 4E 75 0D 0A   : C3
21E0   50 52 49 4E 54 45 52 20   : 44
21E8   44 52 49 56 45 52 20 66   : 52
21F0   6F 72 20 58 36 38 30 30   : 27
21F8   30 20 76 65 72 73 69 6F   : E8
-----------------------------------------
SUM:   86 82 93 66 20 09 C4 E5   E910

2200   6E 20 31 2E 32 33 20 20   : 92
```

```
2208  62 79 20 4E 61 74 63 68  : E9
2210  73 6F 66 74 0D 0A 50 52  : 75
2218  4E 2F 4C 50 54 82 CC 83  : 3E
2220  74 83 40 83 43 83 8B 96  : A1
2228  BC 82 C5 83 76 83 8A 83  : 8C
2230  93 83 5E 82 C9 88 F3 8E  : C8
2238  9A 89 C2 94 5C 82 C5 82  : 9E
2240  B7 81 42 81 69 50 50 50  : 54
2248  82 C5 31 36 2F 32 34 83  : C6
2250  68 83 62 83 67 90 D8 82  : 21
2258  E8 91 D6 82 A6 93 99 81  : 24
2260  6A 0D 0A 00 0D 0A 83 70  : 8B
2268  83 89 83 81 81 5B 83 5E  : CD
2270  82 C9 8C EB 82 E8 82 AA  : 58
2278  82 A0 82 E8 82 DC 82 B7  : 23
--------------------------------------
SUM:  68 A1 6E 6C 09 11 6B 8B   6875

2280  81 42 00 00 00 00 00 06  : C9
2288  00 04 00 10 00 04 00 04  : 1C
2290  00 04 00 04 00 04 00 04  : 10
2298  00 04 00 04 00 04 00 04  : 10
22A0  00 04 00 04 00 04 00 06  : 12
22A8  00 04 00 10 00 04 00 04  : 1C
22B0  00 04 00 04 00 04 00 04  : 10
22B8  00 04 00 04 00 04 00 04  : 10
22C0  00 04 00 04 00 0A 00 04  : 16
22C8  00 10 00 04 00 04 00 04  : 1C
22D0  00 04 00 04 00 04 00 04  : 10
22D8  00 04 00 04 00 04 00 04  : 10
22E0  00 04 00 06 00 0C 00 06  : 1C
22E8  00 30 00 0C 00 06 00 08  : 4A
22F0  00 0C 00 06 00 8E 00 06  : A6
22F8  00 06 00 04 00 06 00 04  : 14
--------------------------------------
SUM:  81 C0 00 60 00 D8 00 4C   0A66

2300  00 06 00 06 00 06 00 06  : 18
```

```
2308  00 0A 00 0A 00 06 00 04  : 1E
2310  00 0A 00 06 00 0A 00 0A  : 24
2318  00 06 00 04 00 06 00 0C  : 1C
2320  00 06 00 04 00 06 00 06  : 16
2328  00 0E 00 06 00 04 00 06  : 1E
2330  00 06 00 04 00 08 00 06  : 18
2338  00 06 00 04 00 08 00 06  : 18
2340  00 06 00 06 00 06 00 06  : 18
2348  00 06 00 06 00 06 00 2C  : 3E
2350  00 22 00 08 00 1E 00 0A  : 52
2358  00 08 00 08 00 0A 00 0A  : 24
2360  00 0A 00 0A 00 0A 00 0A  : 28
2368  00 0A 00 0A 00 5A 00 0C  : 7A
2370  00 18 00 0A 00 08 00 0A  : 34
2378  00 08 00 08 00 0E 00 18  : 36
--------------------------------------
SUM:  00 AA 00 6E 00 E4 00 B6   F5DA

2380  00 08 00 0E 00 20 00 0E  : 44
2388  00 0E 00 08 00 08 00 0A  : 28
2390  00 06 00 36 00 10 00 2A  : 76
2398  00 06 00 0C 00 0C 00 0C  : 2A
23A0  00 08 00 08 00 08 00 08  : 20
23A8  00 22 00 7A 00 08 00 06  : AA
23B0  00 08 00 08 00 0A 00 06  : 20
23B8  00 08 00 18 00 10 00 08  : 38
23C0  00 1A 00 0C 00 08 00 08  : 36
23C8  00 0C 00 08 00 08 00 06  : 22
23D0  00 06 00 06 00 06 00 06  : 18
23D8  00 0A 00 06 00 10 00 08  : 28
23E0  00 0A 00 08 00 06 00 0C  : 24
23E8  00 0A 00 08 00 06 00 08  : 20
23F0  00 08 00 06 00 08 00 06  : 1C
23F8  00 16 00 08 00 08 00 0C  : 32
--------------------------------------
SUM:  00 C4 00 38 00 B0 00 AC   B8D0

2400  00 08 00 06 00 06 00 06  : 1A
```

```
2408  00 08 00 06 00 10 00 06  : 24
2410  00 0C 00 10 00 14 00 08  : 38
2418  00 0E 00 0A 00 08 00 16  : 36
2420  00 1E 00 22 00 30 00 0A  : 7A
2428  00 10 00 1C 00 0C 00 14  : 4C
2430  00 06 00 24 00 0A 00 06  : 3A
2438  00 06 00 12 00 06 00 28  : 46
2440  00 12 00 08 00 24 00 06  : 44
2448  00 1E 00 0A 00 06 00 06  : 34
2450  00 0C 00 0C 00 0C 00 0A  : 2E
2458  00 14 00 06 00 06 00 30  : 50
2460  00 12 00 18 00 06 00 0C  : 3C
2468  00 0A 00 06 00 0E 00 14  : 32
2470  00 0A 00 06 00 0E 00 18  : 36
2478  00 0A 00 06 00 10 00 06  : 26
--------------------------------------
SUM:  00 E4 00 E8 00 EC 00 FA   6A2B

2480  00 08 00 16 00 06 00 06  : 2A
2488  00 10 00 06 00 10 00 10  : 36
2490  00 56 00 0A 00 10 00 06  : 76
2498  00 08 00 3C 00 0A 00 06  : 54
24A0  00 1A 00 16 00 08 00 06  : 3E
24A8  00 06 00 04 00 06 00 04  : 14
24B0  00 18 00 06 00 06 00 06  : 2A
24B8  00 04 00 06 00 04 00 06  : 14
24C0  00 0A 00 14 00 08 00 3C  : 62
24C8  00 06 00 28 00 10 00 30  : 6E
24D0  00 16 00 04 00 06 00 04  : 24
24D8  00 06 00 06 00 06 00 26  : 38
24E0  00 22 11 E0 00 04 00 96  : AD
24E8  00 0A 00 0C 00 0E 00 0E  : 32
24F0  00 0A 00 10 00 00 00 00  : 1A
24F8  00 00 00 00 00 00 00 00  : 00
--------------------------------------
SUM:  00 14 11 CA 00 7E 00 72   AE06
```

リスト2 NPRNDRV.SYSソースリスト

```
=================  NPRNDRV.S  =================
  1: ***********************************************************************
  2: *
  3: *  プリンタドライバ (PC-8821/8822用)
  4: *
  5: *  Copyright 1989/1/2(月)  Ver 1.00 16ドットモードのみ   by Natchsoft
  6: *                1/26(木)  Ver 1.23
  7: *
  8: ***********************************************************************
  9:
 10:    text
 11:
 12: comcod     equ     2
 13: errlow     equ     3
 14: erhigh     equ     4
 15: devend     equ     14
 16: parptr     equ     18
 17: bufadr     equ     14
 18: buflen     equ     18
 19:
 20: tab        equ     $9
 21: lf         equ     $a
 22: ff         equ     $c
 23: cr         equ     $d
 24: so         equ     $e
 25: si         equ     $f
 26: eof        equ     $1a
 27: esc        equ     $1b
 28: fs         equ     $1c
 29:
 30: _PRINT     equ     $ff09
 31:
 32: _B_BITS    equ     $4
 33: _CRTMOD    equ     $10
 34: _FNTGET    equ     $19
 35: _SNSPRN    equ     $3d
 36: _OUTLPT    equ     $3e
 37: _OUTPRN    equ     $3f
 38: _B_INTV    equ     $80
 39: _APAGE     equ     $b1
 40:
 41: _COPY      equ     $2c       *TRAP12 割り込みベクタ番号
 42:
 43: T_VRAM     equ     $e00000 *テキストビデオラム先頭アドレス
 44: G_VRAM     equ     $c00000 *グラフィックビデオラム先頭アドレス
 45: TIMEOUT    equ     $ed003c *プリンタタイムアウト時間
 46:
 47: INTMSK     equ     $e88013 *MFP 割り込みマスク・レジスタA
 48: PRDATA     equ     $e8c001 *プリンタ・データ
 49: STROBE     equ     $e8c003 *プリンタ・ストローブ
 50: PRBUSY     equ     $e9c001 *プリンタ・ビジー
 51: BRK_ON     equ     1         *ブレイクキーが押された時 _B_BITS が返す値はこのビットが
立っている
 52: KEYGRP     equ     $c        *ブレイクキーのキーコードグループ
 53: XFKEYG     equ     $a        *XFキーのキーコードグループ
 54: INT_ON     equ     $18       *キー入力割り込みを受け付ける( INTMSK にセット)
 55:
 56: prntbl:    dc.l    lpttbl
 57:    dc.w    $8000
 58:    dc.l    prnstr
 59:    dc.l    prnent
 60:    dc.b    'PRN     '
 61: prnreq:    dc.l    0
 62: prnjmp:    dc.l    init
 63:    dc.l    notcom
 64:    dc.l    notcom
 65:    dc.l    notcom
 66:    dc.l    notcom
 67:    dc.l    notcom
 68:    dc.l    notcom
 69:    dc.l    notcom
 70:    dc.l    prnout
 71:    dc.l    prnout
 72:    dc.l    prnosn
 73:    dc.l    okcom
 74:    dc.l    notcom
 75:
 76: lpttbl:    dc.l    ppptbl
 77:    dc.w    $8000
 78:    dc.l    lptstr
 79:    dc.l    lptent
 80:    dc.b    'LPT     '
 81: lptreq:    dc.l    0
 82: lptjmp:    dc.l    init
 83:    dc.l    notcom
 84:    dc.l    notcom
 85:    dc.l    notcom
 86:    dc.l    notcom
 87:    dc.l    notcom
 88:    dc.l    notcom
 89:    dc.l    notcom
 90:    dc.l    lptout
 91:    dc.l    lptout
 92:    dc.l    prnosn
 93:    dc.l    okcom
 94:    dc.l    notcom
 95:
 96: ppptbl:    dc.l    -1
 97:    dc.w    $8000
 98:    dc.l    pppstr
 99:    dc.l    pppent
100:    dc.b    'PPP     '
101: pppreq:    dc.l    0
102: pppjmp:    dc.l    init
103:    dc.l    notcom
104:    dc.l    notcom
105:    dc.l    notcom
106:    dc.l    notcom
107:    dc.l    notcom
108:    dc.l    notcom
109:    dc.l    notcom
110:    dc.l    pppout
111:    dc.l    pppout
112:    dc.l    okcom
113:    dc.l    okcom
114:    dc.l    notcom
115:
116: prnstr:    move.l  a5,prnreq
117:    rts
118:
119: prnent:    movem.l d0/a4-a5,-(sp)
120:    move.l  prnreq,a5
121:    lea     prnjmp,a4
122: entry:     moveq.l #0,d0
123:    move.b  comcod(a5),d0
```

▶Oh！Xのなかにもアイドルファンは多いと思うが，僕の一押しのアイドルは吉田真理子である。その次が中山忍。以下順に，佐野量子，宮沢りえ，渡辺満里奈，Wink(特に鈴木早智子)，小川範子，西田ひかる，本多理沙，西村知美，菊池桃子……などです。編集部でこれらすべての名前を知ってる人は僕と同類です。　小杉　文武（17）静岡県

```
124:   add.w    d0,d0
125:   add.w    d0,d0
126:   move.l   0(a4,d0.w),a4
127:   jsr      (a4)
128:   move.b   d0,errlow(a5)
129:   lsr.w    #8,d0
130:   move.b   d0,erhigh(a5)
131:   movem.l  (sp)+,d0/a4-a5
132:   rts
133:
134: notcom:    move.w   #$5003,d0
135:   rts
136:
137: okcom:     clr.w    d0
138:   rts
139:
140: lptstr:    move.l   a5,lptreq
141:   rts
142:
143: lptent:    movem.l d0/a4-a5,-(sp)
144:   move.l   lptreq,a5
145:   lea      lptjmp,a4
146:   bra      entry
147:
148: pppstr:    move.l   a5,pppreq
149:   rts
150:
151: pppent:    movem.l d0/a4-a5,-(sp)
152:   move.l   pppreq,a5
153:   lea      pppjmp,a4
154:   bra      entry
155:
156: prnout:    movem.l d1/d3/a1,-(sp)
157:   move.l   buflen(a5),d3
158:   beq      prnend
159:   move.l   bufadr(a5),a1
160: prnlop:    moveq.l #0,d1
161:   move.b   (a1)+,d1
162:   moveq.l  #_OUTPRN,d0
163:   trap     #15
164:   subq.l   #1,d3
165:   bne      prnlop
166: prnend:    movem.l (sp)+,d1/d3/a1
167:   bra      okcom
168:
169: lptout:    movem.l d1/d3/a1,-(sp)
170:   move.l   buflen(a5),d3
171:   beq      prnend
172:   move.l   bufadr(a5),a1
173: lptlop:    moveq.l #0,d1
174:   move.b   (a1)+,d1
175:   moveq.l  #_OUTLPT,d0
176:   trap     #15
177:   subq.l   #1,d3
178:   bne      lptlop
179:   bra      prnend
180:
181: pppout:    movem.l d1/d3/a1,-(sp)
182:   move.l   buflen(a5),d3
183:   beq      prnend
184:   move.l   bufadr(a5),a1
185:   move.b   (a1),d0
186:   cmp.b    #'0',d0 *バッファ初期化
187:   beq      pbfclr
188:   cmp.b    #'1',d0 *16ドットモードにする
189:   beq      p16md
190:   cmp.b    #'2',d0 *24ドットモードにする
191:   beq      p24md
192:   bclr.l   #5,d0   *大文字にする
193:   cmp.b    #'E',d0 *_OUTLPT出力をエミュレートモードにする
194:   beq      lptemu
195:   cmp.b    #'S',d0 *_OUTLPT出力をスモールエミュレートモードにする
196:   beq      lptsem
197:   cmp.b    #'N',d0 *_OUTLPT出力を正常にする
198:   beq      lptnrm
199:   cmp.b    #'C',d0 *_OUTPRN／_OUTLPT出力をプリンタ出力チェックモードにする
(他のコマントを送ると解除)
200:   beq      chkpst
201:   bra      prnend
202:
203: pbfclr:    clr.b    chkpfg
204:   clr.w    x
205:   move.l   #buff,bimptr
206:   move.l   #buff2,bimpt2
207:   clr.w    bimnum
208:   clr.b    dotflg
209:   clr.b    baiflg
210:   clr.b    hcflag
211:   bra      prnend
212:
213: p16md:     tst.b    p24fg
214:   beq      prnend
215:   clr.b    p24fg
216:   move.b   leng16,length
217: p16md2:    bsr      tstcr
218:   clr.b    chkpfg
219:   clr.b    sjis_1
220:   bra      prnend
221:
222: p24md:     tst.b    p24fg
223:   bne      prnend
224:   st.b     p24fg
225:   move.b   leng24,length
226:   lea      katacd,a1
227:   bsr      lpout
228:   bra      p16md2
229:
230: lptemu:    st.b     emuflg
231:   move.w   limstd,limnum
232: lptem2:    clr.b    chkpfg
233:   lea      katacd,a1
234:   bsr      lpout
235:   bra      prinit
236:
237: lptsem:    move.b   #1,emuflg
238:   move.w   limsml,limnum
239:   st.b     p24fg
240:   move.b   leng24,length
241:   bra      lptem2
242:
243: lptnrm:    clr.b    chkpfg
244:   clr.b    emuflg
245:   move.w   limstd,limnum
246:   bra      prinit
247:
248: chkpst:    st.b     chkpfg
249: prinit:    clr.l    bimctr
250:   clr.w    tflag
251:   clr.w    iniflg
252:   clr.b    abposf
253:   clr.w    pagelf
254:   clr.b    baiflg
255:   clr.b    dotflg
256:   move.b   stlfw,d1
257:   bsr      lfwset
258:   bra      prnend
259:
260: prnosn:    moveq.l #_SNSPRN,d0
261:   trap     #15
262:   tst.b    d0
263:   bne      okcom
264:   moveq.l  #1,d0
265:   rts
266:
267: _B_LPT:    move.l   TIMEOUT,d0
268: _B_LPL:    btst.b   #5,PRBUSY
269:   bne      _B_LP2
270:   tst.b    musifg
271:   bne      _B_LPE
272:   subq.l   #1,d0
273:   bne      _B_LPL
274:   movem.l  d1-d7/a0-a6,-(sp)
275:   move.w   #$7009,d7
276:   trap     #14                  *プリンタオフライン
277:   move.w   d7,d0
278:   movem.l  (sp)+,d1-d7/a0-a6
279:   cmp.b    #2,d0
280:   bne      _B_LPT               *再実行の時
281:   st.b     musifg               *無視の時
282: _B_LPE:    rts
283:
284: _B_LP2:    clr.b    musifg
285:   move.b   d1,PRDATA            *プリンタ出力データ
286:   clr.b    STROBE               *プリンタストローブ
287:   bsr      wait                 *待時間
288:   move.b   #1,STROBE            *プリンタストローブ
289: wait:      rts
290:
291: outlpt:    tst.b    chkpfg
292:   bne      chkprn
293:   tst.b    prnflg
294:   bne      outlp2
295:   tst.b    emuflg
296:   beq      _B_LPT
297: outlp2:    tst.w    bimctr
298:   bne      bimexc
299:   tst.b    tflag
300:   bne      txxnxt
301:   tst.b    bimflg
302:   bne      bimnxt
303:   tst.b    iniflg
304:   bne      inil
305:   tst.b    reposf
306:   bne      rpsnxt
307:   tst.b    abposf
308:   bne      apsnxt
309:   tst.b    pagelf
310:   bne      pgnext
311:   tst.b    escflg
312:   bne      escnxt
313:   tst.b    fsflag
314:   bne      fsnext
315:   cmp.b    #' ',d1
316:   bcc      notctl
317:   cmp.b    #lf,d1
318:   beq      lfout
319:   cmp.b    #ff,d1
320:   beq      ffout
321:   cmp.b    #cr,d1
322:   beq      crexc
323:   cmp.b    #tab,d1
324:   beq      tabout
325:   cmp.b    #esc,d1
326:   beq      escl
327:   cmp.b    #fs,d1
328:   beq      fsl
329:   cmp.b    #so,d1
330:   beq      baiset
331:   cmp.b    #si,d1
332:   beq      baiclr
333:   cmp.b    #eof,d1
334:   beq      lfout
335:   bra      chkprn
336: notctl:    tst.b    prnflg
337:   bne      lprint
338:   cmp.b    #1,emuflg
339:   beq      lprint
340:   rts                    *楽譜印字時、右端が切れてしまうので仕方がない
341:
342: crexlf:    bsr      crexc
343:   bra      lfout
344:
345: crexc:     movem.l d1-d3/a1,-(sp)
346:   cmp.b    #1,emuflg
347:   beq      crsml
348:   tst.b    dotflg
349:   beq      crchar
350:   tst.b    emuflg
351:   bne      cremu
352: crchar:    tst.b    p24fg
353:   bne      cr24
354:   move.w   bimnum,d3
```


▶(て)さん，JR西日本の奥出雲の銘水シリーズ「二十世紀梨ドリンク」をぜひどーぞ。JR西日本の駅で入手できます。九州でも小倉駅の北口ではありました。いま，私の目の前にはその「二十世紀梨ドリンク」があります。それでは先立つ不幸をお許しください。ゴックン。ぎえーっ！

升井　晋也（20）福岡県

```
355:   beq      crexce
356: crexc2:    lea     lfstcd,a1
357:   bsr      lpout
358:   bsr      dotout
359:   lea      buff,a1
360:   move.l   d3,d2
361:   add.w    d2,d2
362:   bsr      lpoutx
363: crexce:    bsr     crout
364:   movem.l  (sp)+,d1-d3/a1
365:   rts
366:
367: cr24:      move.w  bimnum,d3
368:   beq      cr24e
369:   lea      lf8cd,a1
370:   bsr      lpout
371:   bsr      dotout
372:   lea      buff2,a1
373:   move.w   d3,d2
374:   subq.w   #1,d2
375: cr241:     move.b  (a1)+,d1
376:   bsr      _B_LPT
377:   clr.b    d1
378:   bsr      _B_LPT
379:   dbra     d2,cr241
380:   bsr      crlf
381:   bra      crexc2
382:
383: cr24e:     lea     lf8cd,a1
384:   bsr      lpout
385:   bsr      crlf
386:   lea      lfstcd,a1
387:   bsr      lpout
388:   bra      crexce
389:
390: cremu:     cmp.b   #1,emuflg
391:   bne      cr24
392: crsml:     move.w  bimnum,d3           *スモールエミュレートモード時
393:   beq      crexce
394:   move.l   bimptr,a1
395:   clr.w    (a1)
396:   move.l   a2,-(sp)
397:   lea      buff,a1
398:   lea      buff2,a2
399:   subq.w   #1,d3
400: crsml2:    moveq.l #7,d1
401:   move.w   (a1),d2
402:   rol.w    #8,d2
403: crsml3:    add.w   d2,d2
404:   addx.w   d0,d0
405:   cmp.w    #2,d1
406:   bne      crsml4
407:   move.b   (a2)+,d2
408:   rol.w    #8,d2
409: crsml4:    add.w   d2,d2
410:   addx.w   d0,d0
411:   add.w    d2,d2
412:   bcc      crsml5
413:   or.w     #1,d0
414: crsml5:    dbra    d1,crsml3
415:   rol.w    #8,d0
416:   move.w   d0,(a1)+
417:   dbra     d3,crsml2
418:   moveq.l  #0,d0
419:   move.w   bimnum,d0
420:   subq.w   #1,d0
421:   divu     #3,d0
422:   moveq.l  #1,d3
423:   add.w    d0,d3
424:   lea      buff,a1
425:   move.l   a1,a2
426: crsml6:    move.w  (a2)+,(a1)+
427:   move.w   (a2)+,d1
428:   or.w     (a2)+,d1
429:   move.w   d1,(a1)+
430:   dbra     d0,crsml6
431:   move.l   (sp)+,a2
432:   swap     d0
433:   add.w    d3,d3
434:   tst.w    d0
435:   bne      crexc2
436:   subq.w   #1,d3
437:   bra      crexc2
438:
439: tabout:    movem.l d1-d2/a1-a2,-(sp)
440:   move.b   baiflg,d2
441:   clr.b    baiflg
442: tabotl:    moveq.l #' ',d1
443:   bsr      lprint
444:   move.b   x,d0
445:   and.b    #7,d0
446:   bne      tabotl
447:   move.b   d2,baiflg
448:   movem.l  (sp)+,d1-d2/a1-a2
449:   rts
450:
451: escl:      st.b    escflg
452:   rts
453:
454: fsl:       st.b    fsflag
455:   rts
456:
457: baiset:    st.b    baiflg
458:   rts
459:
460: baiclr:    clr.b   baiflg
461:   rts
462:
463: fsnext:    clr.b   fsflag
464:   cmp.b    #'p',d1
465:   beq      baiset
466:   cmp.b    #'q',d1
467:   beq      baiclr
468:   move.w   d1,-(sp)          *FS の後が p または q でない時
469:   move.b   #fs,d1
470:   bsr      chkprn
```

```
471:   move.w   (sp)+,d1
472:   bra      chkprn
473:
474: escnxt:    clr.b   escflg
475:   cmp.b    #'%',d1            *N/120インチ改行幅設定
476:   beq      txx1
477:   cmp.b    #'J',d1            *ビットイメージ24ドット
478:   beq      bim1
479:   cmp.b    #'c',d1            *プリンタの初期化のはずだが、ここではコードを読み飛ばすだけ
480:   beq      ini1
481:   cmp.b    #'p',d1            *紙切れ時の設定のはずだが、ここではコードを読み飛ばすだけ
482:   beq      ini1
483:   cmp.b    #'¥',d1            *ドット単位印字位置相対指定
484:   beq      repos1
485:   cmp.b    #$10,d1            *ドット単位印字位置絶対指定
486:   beq      abpos1
487:   cmp.b    #'F',d1            *コードを読み飛ばすだけ
488:   beq      pagel1
489:   cmp.b    #'6',d1            *1/6インチ改行
490:   beq      lfw6
491:   cmp.b    #'f',d1            *PC-8821/8822 正方向改行
492:   beq      escnx2
493:   cmp.b    #'r',d1            *PC-8821/8822 逆方向改行
494:   beq      escnx2
495:   cmp.b    #']',d1            *両方向印字
496:   beq      escnx2
497:   cmp.b    #'>',d1            *片方向印字
498:   beq      escnx2
499:   cmp.b    #'5',d1            *ページ先頭指定(コードを読み飛ばすだけ)
500:   beq      escrts
501:
502:   move.w   d1,-(sp)           *未定義コード
503:   move.b   #esc,d1
504:   bsr      chkprn
505:   move.w   (sp)+,d1
506:   bra      chkprn
507:
508: escrts:    rts
509:
510: escnx2:    move.w  d1,-(sp)
511:   move.b   #esc,d1
512:   bsr      _B_LPT
513:   move.w   (sp)+,d1
514:   bra      _B_LPT
515:
516: txx1:      st.b    tflag
517:   rts
518:
519: bim1:      st.b    bimflg
520:   st.b     dotflg
521:   rts
522:
523: ini1:      seq.b   iniflg
524:   rts
525:
526: repos1:    st.b    reposf
527:   rts
528:
529: abpos1:    move.b  #4,abposf
530:   clr.w    abposd
531:   rts
532:
533: pagel1:    st.b    pagelf
534:   rts
535:
536: lfw6:      move.w  d1,-(sp)
537:   move.b   #20,d1
538:   bsr      lfwset
539:   move.w   (sp)+,d1
540:   rts
541:
542: txxnxt:    cmp.b   #2,tflag
543:   beq      txxset
544:   cmp.b    #'9',d1
545:   bne      tnot_9
546:   move.b   #2,tflag
547:   rts
548:
549: tnot_9:    clr.b   tflag      *ESC % の後が 9 でなかった場合
550:   move.l   d1,-(sp)
551:   moveq.l  #esc,d1
552:   bsr      chkprn
553:   moveq.l  #'%',d1
554:   bsr      chkprn
555:   move.l   (sp)+,d1
556:   bra      chkprn
557:
558: txxset:    clr.b   tflag
559:   bra      lfwset
560:
561: bimnxt:    cmp.b   #2,bimflg
562:   beq      bimset
563:   move.b   d1,bimdot
564:   move.b   #2,bimflg
565:   rts
566:
567: bimset:    movem.l d1-d2,-(sp)
568:   clr.b    bimflg
569:   rol.w    #8,d1
570:   move.b   bimdot,d1
571:   rol.w    #8,d1
572:   move.w   d1,bimctr
573:   add.w    d1,bimnum
574:   move.w   d1,bimnu2
575:   clr.b    bimexf
576:   move.w   limnum,d2
577:   move.w   bimnum,d1
578:   sub.w    d2,d1
579:   bls      bimste
580:   sub.w    d1,bimnu2
581:   move.w   d2,bimnum
582: bimste:    movem.l (sp)+,d1-d2
583:   rts
584:
585: bimexc:    movem.l d1-d3/a1,-(sp)
586:   tst.b    bimexf
```

▶11月から，我が尾張旭市もついに市外局番が4桁に。よかった，これで学校の友達にバカにされなくてすむ。　水野　義則（17）愛知県

```
587:    bne      bimex2
588:    tst.w    bimnu2
589:    beq      doverl
590:    bsr      bitrev
591:    move.l   bimpt2,a1
592:    move.b   d0,(a1)+
593:    move.l   a1,bimpt2
594: doverl:     addq.b  #1,bimexf
595: bimend:     movem.l (sp)+,d1-d3/a1
596:    rts
597:
598: bimex2:     tst.w   bimnu2
599:    beq      dover2
600:    bsr      bitrev
601:    move.l   bimptr,a1
602:    move.b   d0,(a1)+
603:    move.l   a1,bimptr
604: dover2:     addq.b  #1,bimexf
605:    cmp.b    #2,bimexf
606:    beq      bimend
607:    clr.b    bimexf
608:    tst.w    bimnu2
609:    beq      dover3
610:    subq.w   #1,bimnu2
611: dover3:     subq.w  #1,bimctr
612:    bra      bimend
613:
614: bitrev:     moveq.l #7,d2       *parameter d1 , out d0 , break d0-d2
615: bitrvl:     lsr.b   #1,d1       *bit reverse
616:    roxl.b   #1,d0
617:    dbra     d2,bitrvl
618:    rts
619:
620: rpsnxt:     cmp.b   #2,reposf
621:    beq      rpsset
622:    move.b   d1,rpslow
623:    move.b   #2,reposf
624:    rts
625:
626: rpsset:     movem.l d1/a1-a2,-(sp)
627:    clr.b    reposf
628:    rol.w    #8,d1
629:    move.b   rpslow,d1
630: apsset:     add.w   d1,bimnum
631:    move.w   bimnum,d0
632:    cmp.w    limnum,d0
633:    bhi      psover
634:    move.l   bimptr,a1
635:    move.l   bimpt2,a2
636:    bra      rpsst3
637: rpsst2:     clr.w   (a1)+
638:    clr.b    (a2)+
639: rpsst3:     dbra    d1,rpsst2
640:    move.l   a1,bimptr
641:    move.l   a2,bimpt2
642: posend:     movem.l (sp)+,d1/a1-a2
643:    rts
644:
645: psover:     sub.w   d1,bimnum
646:    bsr      crexlf
647:    bra      posend
648:
649: apsnxt:     movem.l d1/a1-a2,-(sp)
650:    move.w   abposd,d0
651:    mulu     #10,d0
652:    and.w    #$ff,d1
653:    sub.b    #'0',d1
654:    add.w    d0,d1
655:    subq.b   #1,abposf
656:    beq      apsset
657:    move.w   d1,abposd
658:    movem.l  (sp)+,d1/a1-a2
659:    rts
660:
661: pgnext:     cmp.b   #2,pagelf
662:    beq      pagset
663:    move.b   #2,pagelf
664:    rts
665:
666: pagset:     clr.b   pagelf
667:    rts
668:
669: chkprn:     movem.l d1-d2/a1,-(sp)
670:    move.l   d1,d2
671:    moveq.l  #esc,d1
672:    bsr      _B_LPT
673:    moveq.l  #'H',d1  *HDバイカ
674:    tst.b    prnflg
675:    bne      chkprl
676:    moveq.l  #'N',d1  *HSバイカ
677: chkprl:     bsr     _B_LPT
678:    move.l   d2,d1
679:    lsr.b    #4,d1
680:    bsr      chksub
681:    move.l   d2,d1
682:    and.b    #$f,d1
683:    bsr      chksub
684:    moveq.l  #7,d2
685:    lea      dspccd,a1
686:    bsr      lpoutx
687:    movem.l  (sp)+,d1-d2/a1
688:    rts
689:
690: chksub:     cmp.b   #10,d1
691:    bcs.s    chksb2
692:    add.b    #'A'-'9'-1,d1
693: chksb2:     add.b   #'0',d1
694:    bra      _B_LPT
695:
696: outprn:     move.l  d1,-(sp)
697:    st.b     prnflg
698:    ror.w    #8,d1
699:    cmp.b    #$80,d1
700:    bcs      outpr3
701:    cmp.b    #$a0,d1
702:    bcs      outpr2
703:    cmp.b    #$e0,d1
704:    bcs      outpr3
705: outpr2:     and.l   #$ff,d1
706:    bsr      outlpt
707: outpr3:     move.l  (sp),d1
708:    and.l    #$ff,d1
709:    bsr      outlpt
710:    clr.b    prnflg
711:    move.l   (sp)+,d1
712:    rts
713:
714: lprint:     tst.b   p24fg
715:    beq      lpri2
716:    swap     d1
717:    move.w   #$c,d1   *24 dot font
718:    swap     d1
719: lpri2:      tst.b   sjis_1
720:    bne      kan2
721:    cmp.b    #$80,d1
722:    bcs      ank
723:    cmp.b    #$a0,d1
724:    bcs      kan1
725:    cmp.b    #$e0,d1
726:    bcs      ank
727:
728: kan1:       move.b  d1,sjis_1
729:    rts
730:
731: ank:        movem.l d2-d4/a0-a3,-(sp)
732:    lea      fontbf,a1
733:    moveq.l  #_FNTGET,d0
734:    trap     #15
735:    cmp.w    #12,(a1)
736:    beq      ank24
737: ank16:      bsr     bimchk
738:    move.l   bimptr,a2
739:    move.l   bimpt2,a3
740:    moveq.l  #7,d2
741: ank2:       move.l  a1,a0
742:    moveq.l  #1,d3
743: ank3:       moveq.l #7,d4
744: ank4:       lsr.b   #1,d1
745:    btst.b   d2,(a0)+
746:    beq      ank5
747:    or.b     #$80,d1
748: ank5:       dbra    d4,ank4
749:    move.b   d1,(a2)+
750:    dbra     d3,ank3
751:    clr.b    (a3)+
752:    addq.w   #1,bimnum
753:    dbra     d2,ank2
754: ankend:     move.l  a2,bimptr
755:    move.l   a3,bimpt2
756:    addq.b   #1,x
757:    movem.l  (sp)+,d2-d4/a0-a3
758:    bra      prexit
759:
760: ank24:      bsr     bimchk
761:    move.l   bimptr,a2
762:    move.l   bimpt2,a3
763:    moveq.l  #11,d2
764: ank242:     move.l  a1,a0
765:    moveq.l  #7,d3
766: ank243:     lsl.w   (a0)+
767:    roxr.b   #1,d1
768:    dbra     d3,ank243
769:    move.b   d1,(a3)+
770:    moveq.l  #1,d4
771: ank244:     moveq.l #7,d3
772: ank245:     lsl.w   (a0)+
773:    roxr.b   #1,d1
774:    dbra     d3,ank245
775:    move.b   d1,(a2)+
776:    dbra     d4,ank244
777:    addq.w   #1,bimnum
778:    dbra     d2,ank242
779:    bra      ankend
780:
781: kan2:       movem.l d2-d4/a0-a3,-(sp)
782:    rol.w    #8,d1
783:    move.b   sjis_1,d1
784:    rol.w    #8,d1
785:    lea      fontbf,a1
786:    moveq.l  #_FNTGET,d0
787:    trap     #15
788:    move.w   (a1),d0
789:    cmp.w    #8,d0
790:    beq      ank16
791:    cmp.w    #12,d0
792:    beq      ank24
793:    cmp.w    #24,d0
794:    beq      kan24
795:    bsr      bimchk
796:    move.l   bimptr,a2
797:    move.l   bimpt2,a3
798:    moveq.l  #15,d2
799: kanji2:     move.l  a1,a0
800:    moveq.l  #1,d3
801: kanji3:     moveq.l #7,d4
802: kanji4:     lsl.w   (a0)+
803:    roxr.b   #1,d1
804:    dbra     d4,kanji4
805:    move.b   d1,(a2)+
806:    dbra     d3,kanji3
807:    clr.b    (a3)+
808:    addq.w   #1,bimnum
809:    dbra     d2,kanji2
810: kanend:     move.l  a2,bimptr
811:    move.l   a3,bimpt2
812:    addq.b   #2,x
813:    movem.l  (sp)+,d2-d4/a0-a3
814: prexit:     cmp.b   #1,emuflg
815:    bne      prext2
816:    bsr      dot6sp
817: prext2:     tst.b   baiflg
818:    beq      prext3
```

▶「DōGA・CGAシステム」期待してますよー。かまたゆたかさん，ガンバッテください。
太田 京一（20）茨城県

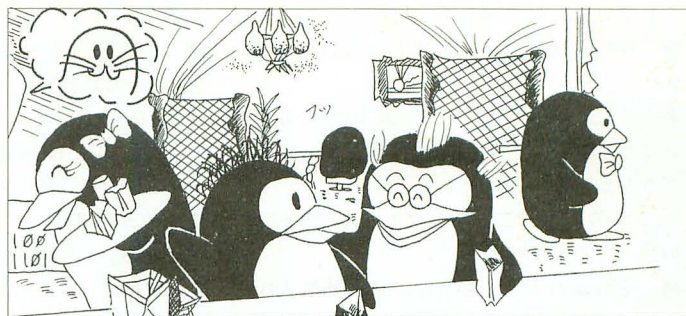
```
819:    bsr     baiexc
820: prext3:    clr.b   sjis_1
821:    rts
822:
823: kan24:     bsr     bimchk
824:    lea     fntbf2,a2
825:    moveq.l #23,d1
826: kan241:    move.b  (a1)+,(a2)+
827:    move.b  (a1)+,(a2)+
828:    move.b  (a1)+,(a2)+
829:    addq.l  #1,a2
830:    dbra    d1,kan241
831:    lea     fntbf2,a1
832:    move.l  bimptr,a2
833:    move.l  bimpt2,a3
834:    moveq.l #23,d2
835: kan242:    move.l  a1,a0
836:    moveq.l #7,d3
837: kan243:    move.l  (a0),d0
838:    add.l   d0,d0
839:    move.l  d0,(a0)+
840:    roxr.b  #1,d1
841:    dbra    d3,kan243
842:    move.b  d1,(a3)+
843:    moveq.l #1,d4
844: kan244:    moveq.l #7,d3
845: kan245:    move.l  (a0),d0
846:    add.l   d0,d0
847:    move.l  d0,(a0)+
848:    roxr.b  #1,d1
849:    dbra    d3,kan245
850:    move.b  d1,(a2)+
851:    dbra    d4,kan244
852:    addq.w  #1,bimnum
853:    dbra    d2,kan242
854:    bra     kanend
855:
856: bimchk:    bsr     bimcsb
857:    cmp.b   #1,emuflg
858:    bne     bimck2
859:    cmp.w   #16+12,(a1)
860:    bcs     bimck2
861:    bsr     dot6sp
862: bimck2:    addq.l  #4,a1
863:    rts
864:
865: dot6sp:    movem.l d1/a1-a2,-(sp)
866:    move.l  bimptr,a1
867:    move.l  bimpt2,a2
868:    moveq.l #2,d1
869: dot6s2:    clr.l   (a1)+
870:    clr.w   (a2)+
871:    addq.w  #2,bimnum
872:    dbra    d1,dot6s2
873:    move.l  a1,bimptr
874:    move.l  a2,bimpt2
875:    movem.l (sp)+,d1/a1-a2
876:    rts
877:
878: bimcsb:    cmp.b   #1,emuflg
879:    bne     bimcs2
880:    addq.w  #6,(a1)
881:    cmp.w   #16+6,(a1)
882:    bcs     bimcs2
883:    addq.w  #6,(a1)
884: bimcs2:    move.w  (a1),d0
885:    tst.b   baiflg
886:    beq     bimcs3
887:    add.w   d0,d0
888: bimcs3:    add.w   bimnum,d0
889:    cmp.w   limnum,d0
890:    bhi     crexlf
891:    moveq.l #1,d0
892:    cmp.b   #1,emuflg
893:    bne     bimcs4
894:    cmp.w   #16+12,(a1)
895:    bcs     bimcs6
896:    bra     bimcs5
897: bimcs4:    cmp.w   #16,(a1)
898:    bcs     bimcs6
899: bimcs5:    moveq.l #2,d0
900: bimcs6:    tst.b   baiflg
901:    beq     bimcs7
902:    add.b   d0,d0
903: bimcs7:    add.b   x,d0
904:    cmp.b   length,d0
905:    bhi     crexlf
906:    rts
907:
908: baiexc:    movem.l a1-a4,-(sp)
909:    addq.b  #1,x
910:    move.w  fontbf,d0
911:    cmp.b   #1,emuflg
912:    bne     baiex2
913:    cmp.w   #16+12,d0
914:    bcs     baiex4
915:    bra     baiex3
916: baiex2:    cmp.w   #16,d0
917:    bcs     baiex4
918: baiex3:    addq.b  #1,x
919: baiex4:    move.l  bimptr,a1
920:    move.l  bimpt2,a2
921:    move.l  a1,a3
922:    move.l  a2,a4
923:    add.w   d0,a1
924:    add.w   d0,a1
925:    add.w   d0,a2
926:    move.l  a1,bimptr
927:    move.l  a2,bimpt2
928:    subq.w  #1,d0
929: baiex5:    move.w  -(a3),-(a1)
930:    move.w  (a3),-(a1)
931:    move.b  -(a4),-(a2)
932:    move.b  (a4),-(a2)
933:    addq.w  #1,bimnum
934:    dbra    d0,baiex5
935:    movem.l (sp)+,a1-a4
936:    rts
937:
938: hcopy:     tas.b   hcflag
939:    bne     hcopyr
940:    movem.l d0-d7/a0-a5,-(sp)
941:    move.l  d0,d7
942:    moveq.l #XFKEYG,d1
943:    moveq.l #_B_BITS,d0
944:    trap    #15
945:    move.l  d0,d6
946:    or.b    #INT_ON,INTMSK    *キー入力を受け付けるようにする
947:    btst.l  #3,d7     *OPT.2
948:    bne     hclf
949:    btst.l  #2,d7     *OPT.1
950:    bne     hcff
951:    btst.l  #0,d7     *SHIFT
952:    bne     hctext
953:    btst.l  #1,d7     *CTRL
954:    bne     hcgrap
955:    btst.l  #5,d6     *XF1
956:    bne     lfrevs
957:    btst.l  #6,d6     *XF2
958:    bne     lfforw
959:    bra     hcopye
960:
961: hctext:    bsr     tstcr
962:    lea     katacd,a1
963:    bsr     lpout
964:    lea     lf16cd,a1
965:    bsr     lpout
966:    move.l  #T_VRAM,a0        *vram start address
967:    move.w  ydot,d3           *y_dot/16-1
968: ycopy:     lea     buff,a2
969:    move.l  a0,a4             *vram pointer
970:    move.w  xdot2,d4          *x_dot/8-1
971: xcopy:     moveq.l #7,d5             *bit number
972: xcopy2:    move.l  a4,a3
973:    moveq.l #1,d6
974: xcopy3:    moveq.l #7,d7
975: xcopy4:    lsr.b   #1,d1
976:    btst.b  d5,(a3)
977:    beq     xcopy5
978:    or.b    #$80,d1           *set bit7
979: xcopy5:    lea     $80(a3),a3
980:    dbra    d7,xcopy4
981:    move.b  d1,(a2)+
982:    dbra    d6,xcopy3
983:    moveq.l #KEYGRP,d1
984:    moveq.l #_B_BITS,d0
985:    trap    #15
986:    btst.l  #BRK_ON,d0        *break key check
987:    bne     hcopye
988:    dbra    d5,xcopy2
989:    addq.l  #1,a4
990:    dbra    d4,xcopy
991:    lea     do768c,a1
992:    bsr     lpout
993:    move.w  xdot,d2
994:    lea     buff,a1
995:    bsr     lpoutx
996:    bsr     crlf
997:    lea     $800(a0),a0
998:    dbra    d3,ycopy
999: hcopye:    movem.l (sp)+,d0-d7/a0-a5
1000:   clr.b   hcflag
1001: hcopyr:   rte
1002:
1003: hcgrap:   moveq.l #-1,d1
1004:   moveq.l #_APAGE,d0
1005:   trap    #15
1006:   tst.l   d0
1007:   bmi     hcopye
1008:   bsr     tstcr
1009:   move.w  gy1024,d3         *y_dot/16-1
1010:   moveq.l #0,d7
1011:   move.w  add960,d7
1012:   lea     do960c,a5
1013:   move.w  gxd960,gxdot
1014:   move.w  gx2960,gxdot2
1015:   moveq.l #-1,d1
1016:   moveq.l #_CRTMOD,d0
1017:   trap    #15
1018:   cmp.b   #16,d0
1019:   beq     gcopy
1020:   cmp.b   #4,d0
1021:   bcs     gcopy
1022:   move.w  gy512,d3          *y_dot/16-1
1023:   move.w  add512,d7
1024:   lea     do512c,a5
1025:   move.w  gxd512,gxdot
1026:   move.w  gx2512,gxdot2
1027: gcopy:    lea     katacd,a1
1028:   bsr     lpout
1029:   lea     lf16cd,a1
1030:   bsr     lpout
1031:   bsr     crlf
1032:   move.l  #G_VRAM,a0        *vram start address
1033: gycopy:   lea     buff,a2
1034:   move.l  a0,a4             *vram pointer
1035:   move.w  gxdot2,d4         *x_dot-1
1036: gxcopy:   move.l  a4,a3
1037:   moveq.l #1,d5
1038: gxcpy2:   moveq.l #7,d6
1039: gxcpy3:   lsr.b   #1,d1
1040:   tst.w   (a3)
1041:   beq     gxcpy4
1042:   or.b    #$80,d1           *set bit7
1043: gxcpy4:   add.l   d7,a3
1044:   dbra    d6,gxcpy3
1045:   move.b  d1,(a2)+
1046:   dbra    d5,gxcpy2
1047:   addq.l  #2,a4
1048:   moveq.l #KEYGRP,d1
1049:   moveq.l #_B_BITS,d0
1050:   trap    #15
```

▶ 7月号から新連載の「泉大介のX-BASIC入門」はとてもわかりやすく，すぐに使えるような実用性のあるプログラムもあって，とてもうれしくて楽しみにしています。これからもX-BASICを理解していきたいので，ぜひがんばってください。 松本 剛 (18) 熊本県

```
1051:    btst.l   #BRK_ON,d0       *break key check
1052:    bne      hcopye
1053:    dbra     d4,gxcopy
1054:    move.l   a5,a1            *bit image out code
1055:    bsr      lpout
1056:    move.w   gxdot,d2         *x_dot*2
1057:    lea      buff,a1
1058:    bsr      lpoutx
1059:    bsr      crlf
1060:    lsl.l    #4,d7
1061:    add.l    d7,a0
1062:    lsr.l    #4,d7
1063:    dbra     d3,gycopy
1064:    bra      hcopye
1065:
1066: hclf:    bsr      lfout
1067:    bra      hcopye
1068:
1069: hcff:    bsr      ffout
1070:    bra      hcopye
1071:
1072: lfrevs:  bsr      tstcr
1073:    lea      lfrevc,a1
1074:    bsr      lpout
1075:    bsr      crlf
1076:    bra      hcopye
1077:
1078: lfforw:  bsr      tstcr
1079:    lea      lfforc,a1
1080:    bsr      lpout
1081:    bsr      crlf
1082:    bra      hcopye
1083:
1084: crlf:    move.l   d1,-(sp)
1085:    bsr      crout
1086:    bra      lfout2
1087: lfout:   move.l   d1,-(sp)
1088:    bsr      tstcr
1089: lfout2:  moveq.l  #lf,d1
1090:    bsr      _B_LPT
1091:    move.l   (sp)+,d1
1092:    rts
1093:
1094: ffout:   move.l   d1,-(sp)
1095:    bsr      tstcr
1096:    moveq.l  #ff,d1
1097:    bsr      _B_LPT
1098:    move.l   (sp)+,d1
1099:    rts
1100:
1101: tstcr:   tst.w    bimnum
1102:    beq      tstcre
1103:    bra      crexc
1104: tstcre:  bra      crout
1105:
1106: crout:   moveq.l  #cr,d1
1107:    bsr      _B_LPT
1108:    move.l   #buff,bimptr
1109:    move.l   #buff2,bimpt2
1110:    clr.w    bimnum
1111:    clr.b    x
1112:    clr.b    dotflg
1113:    rts
1114:
1115: lpout:   move.b   (a1)+,d1         *parameter a1 , break d0-d1/a1
1116:    beq      lpoute
1117:    bsr      _B_LPT
1118:    bra      lpout
1119: lpoute:  rts
1120:
1121: lpoutx:  subq.w   #1,d2
1122: lpotxl:  move.b   (a1)+,d1        *parameter d2,a1 , break d0-d2/a1
1123:    bsr      _B_LPT
1124:    dbra     d2,lpotxl
1125:    rts
1126:
1127: lfwset:  movem.l  d1-d2/a1,-(sp)  *parameter d1 , break d0
1128:    move.l   #lfstcd+2,a1
1129:    moveq.l  #0,d2
1130:    move.b   d1,d2
1131:    divu     #10,d2
1132:    add.b    #'0',d2
1133:    move.b   d2,(a1)+
1134:    swap     d2
1135:    add.b    #'0',d2
1136:    move.b   d2,(a1)
1137:    movem.l  (sp)+,d1-d2/a1
1138:    rts
1139:
1140: dotout:  lea      doutcd,a1        *parameter d3 , break d0-d1/a1
1141:    bsr      lpout
1142:    moveq.l  #0,d1
1143:    move.w   d3,d1
1144: deci4:   divu     #1000,d1         *parameter d1 , break d0-d1
1145:    bsr      dotsub
1146:    divu     #100,d1
1147:    bsr      dotsub
1148:    divu     #10,d1
1149:    bsr      dotsub
1150: dotsub:  add.b    #'0',d1
1151:    bsr      _B_LPT
1152:    clr.w    d1
1153:    swap     d1
1154:    rts
1155:
1156: x:       dc.b     0           *プリンタヘッドの現在のX座標
1157: sjis_1:  dc.b     0           *シフトジスコードの1バイト目
1158: buff:    ds.b     1440*2+2    *16ドット×1440個分のバッファ
1159: buff2:   ds.b     1440        *8ドット×1440個分のバッファ
1160: fontbf:  ds.b     3*24+4      *_FNTGET用のバッファ
1161: fntbf2:  ds.b     4*24        *24ドット時のビット操作を行い易くするためのバッファ
1162: xdot:    dc.w     768*2       *x_dot*2
1163: xdot2:   dc.w     768/8-1     *x_dot/8-1
1164: ydot:    dc.w     512/16-1    *y_dot/16-1
1165: gxdot:   dc.w     960*2       *x_dot*2（下の2つのデータのうちのどちらかが入る）
1166: gxd960:  dc.w     960*2       *↑
1167: gxd512:  dc.w 512*2     *↑
1168: gxdot2:  dc.w 960-1     *x_dot-1（下の2つのデータのうちのどちらかが入る）
1169: gx2960:  dc.w 960-1     *↑
1170: gx2512:  dc.w 512-1     *↑
1171: gy1024:  dc.w 1024/16-1  *y_dot/16-1
1172: gy512:   dc.w 512/16-1 *y_dot/16-1
1173: add960:  dc.w $800     *G_VRAMで16ドット下のドットとのアドレス差 (1024*1024)
1174: add512:  dc.w $400     *G_VRAMで16ドット下のドットとのアドレス差 (512*512)
1175: p24fg:   dc.b -1       *PRNが16ドット出力か24ドット出力か（24ドットの時 -1）
1176: length:  dc.b 80       *現在のモードの1行の文字数
1177: leng16:  dc.b 120      *16ドットモード時の1行の文字数
1178: leng24:  dc.b 80       *24ドットモード時の1行の文字数
1179: limnum:  dc.w 960      *現在のモードの1行のドット数
1180: limstd:  dc.w 960      *通常の1行のドット数
1181: limsml:  dc.w 1440     *スモールエミュレートモード時の1行のドット数
1182:
1183: bimptr:  dc.l buff     *16ドット分のバッファ(buff)へのポインタ
1184: bimpt2:  dc.l buff2    *8ドット分のバッファ(buff2)へのポインタ
1185: bimnum:  dc.w 0    *実際に出力するドット列数
1186: bimnu2:  dc.w 0    *960ドットを越えた分を無視するためのカウンター
1187: bimctr:  dc.w 0    *ビットイメージドット列の残り数
1188: escflg:  dc.b 0    *1つ前のデータが ESC であったことを表す
1189: fsflag:  dc.b 0    *1つ前のデータが FS であったことを表す
1190: tflag:   dc.b 0    *以前のデータが ESC % であったことを表す
1191: bimflg:  dc.b 0    *以前のデータが ESC J であったことを表す
1192: bimdot:  dc.b 0    *ビットイメージドット列数の上位8ビット
1193: bimexf:  dc.b 0    *24ドットを8ドットずつに分けた時上から何番目かを表す
1194: iniflg:  dc.b 0    *以前のデータが ESC c または ESC p であったことを表す
1195: reposf:  dc.b 0    *以前のデータが ESC ¥ であったことを表す
1196: rpslow:  dc.b 0    *相対位置指定の下位8ビット
1197: abposf:  dc.b 0    *以前のデータが ESC POS であったことを表す
1198: abposd:  dc.w 0    *絶対位置指定設定値
1199: pagelf:  dc.b 0    *以前のデータが ESC F であったことを表す
1200: prnflg:  dc.b 0    *_OUTPRN から飛んできていることを表す
1201: emuflg:  dc.b -1    *CZ系プリンターのエミュレートを行うかどうか
1202:
1203: baiflg:  dc.b 0    *倍角モードであるかどうかを表す
1204: dotflg:  dc.b 0    *ビットイメージが含まれるかどうかを表す
1205: hcflag:  dc.b 0    *ハードコピー中かどうかを表す
1206: musifg:  dc.b 0    *オフライン時に無視したかどうか
1207: chkpfg:  dc.b 0    *プリンター出力のチェックを行うモードかどうか
1208: dspccd:  dc.b esc,'S0001',0  *ヘッドを1ドット列進める
1209:
1210: katacd:  dc.b esc,'>',0  *片方向印字
1211: stlfw:   dc.b 20          *改行幅 初期化値
1212: lfstcd:  dc.b esc,'T20',0  *改行幅 設定値 （可変）
1213: lf16cd:  dc.b esc,'T16',0  *改行幅 16/120インチ
1214: lf8cd:   dc.b esc,'T08',0  *改行幅  8/120インチ
1215: lfrevc:  dc.b esc,'r',0  *逆方向改行指定
1216: lfforc:  dc.b esc,'f',0  *正方向改行指定
1217: do768c:  dc.b esc,'I0768',0  *16ドットビットイメージ 768ドット列
1218: do960c:  dc.b esc,'I0960',0  *16ドットビットイメージ 960ドット列
1219: do512c:  dc.b esc,'I0512',0  *16ドットビットイメージ 512ドット列
1220: doutcd:  dc.b esc,'I',0  *16ドットビットイメージ
1221:
1222:   even
1223: init:  movem.l  d1/a0-a1,-(sp)
1224:   move.l  parptr(a5),a1    *パラメータポインタ
1225: sysskp:  move.b  (a1)+,d0
1226:   bne  sysskp
1227: igetac:  move.b  (a1)+,d0
1228:   beq  normal
1229:   cmp.b  #'#',d0
1230:   bne  igetac
1231: nxparg:  move.b  (a1)+,d0
1232:   beq  normal
1233:   cmp.b  #'/',d0
1234:   bne  inierr
1235:   move.b  (a1)+,d0
1236:   and.w  #$df,d0      *大文字にする
1237:   cmp.b  #'T',d0
1238:   bne  inierr
1239: lfset:  moveq.l  #0,d1
1240:   move.b  (a1)+,d1    *改行幅初期設定値
1241:   sub.b  #'0',d1
1242:   mulu  #10,d1
1243:   add.b  (a1)+,d1
1244:   sub.b  #'0',d1
1245:   move.b  d1,stlfw
1246:   bsr  lfwset
1247: normal:  lea  hcopy,a1
1248:   moveq.l  #_COPY,d1    *TRAP12 (COPY KEY)
1249:   moveq.l  #_B_INTV,d0    *_B_INTVCS
1250:   trap  #15
1251:   lea  outlpt,a1
1252:   move.w  #$100+_OUTLPT,d1  *IOCS $3E (_OUTLPT)
1253:   moveq.l  #_B_INTV,d0    *_B_INTVCS
1254:   trap  #15
1255:   lea  outprn,a1
1256:   move.w  #$100+_OUTPRN,d1  *IOCS $3F (_OUTPRN)
1257:   moveq.l  #_B_INTV,d0    *_B_INTVCS
1258:   trap  #15
1259:   pea  mes
1260:   dc.w  _PRINT
1261:   addq.l  #4,sp
1262:   move.l  #init,devend(a5)  *デバイスドライバ終了アドレス
1263:   movem.l  (sp)+,d1/a0-a1
1264:   clr.w  d0
1265:   rts
1266:
1267: inierr:  pea  errmes
1268:   dc.w  _PRINT
1269:   addq.l  #4,sp
1270:   movem.l  (sp)+,d1/a0-a1
1271:   move.w  #$700d,d0
1272:   rts
1273:
1274: mes:  dc.b  13,10
1275:   dc.b 'PRINTER DRIVER for X68000 version 1.23  by Natchsoft',13,10
1276:   dc.b 'PRN/LPTのファイル名でプリンタに印字可能です。（PPPで16/24ドット切り替え等）',13,10,0
1277: errmes:  dc.b  13,10
1278:   dc.b 'パラメータに誤りがあります。',0
1279:
1280:   end
```

マシン語カクテル in Z80's Bar

第3回――謎のゼンジソフト――

シナリオ：金子俊一　西川善司
特別監修：浦川博之
イラスト：山田純二

今夜は一見してどう使うのかわからないシフト命令とローテイト命令について長老の講釈を拝聴しましょう。西川君もプログラムを持参し，今夜は酔いが回った様子。なお，このドラマはフィクションであり実在のソフトハウス，プログラマとは一切関係ありません。

♪カランコロ～ン（ドアの開く音）
マスター(以下M)：いらっしゃい，西川さん。今日も自主休講ですか？
西川善司(以下善)：いえいえ。
M：まさか学校やめちゃったんですか。
善：はははっ，そのとおり。いやぁ，ソフトハウスを作っちゃったんですよ。名づけて「ゼンジソフト」
M：中国は広島生まれの手品師みたいな名前ですね。
善：……。(ひゅーと風が吹く)
M：で，第1号のソフトはなんなんですか。
善：いやぁ。アダルトソフトですよ。はっはっは。(ひゅーと風が吹く)
長老(以下老)：ほほほほ，西川くん。おぬしも上達したのう。しかしまだまだ青いわ。Z80はそんなに簡単に極められるものではないぞ。
善：そ，その声は!?　どこだ!?
老：ちょっと待っておれ，手を洗うからの。(じょじょじょーわぁー，ギィー，とトイレのドアが開いて)
老：はっはっはっ。ここじゃ。しかし西川くん，ソフトハウスを作ったはいいが，まだろくにプログラムも書けんじゃろうが。
善：う゛。
老：まぁよい。今宵も，わしの話をしっかり聞いて精進されよ。ふぉっふぉっふぉっふぉ。

シフト命令の怪

老：ようこくん。西川くんにはスイカソーダ2(注1)，わしにはサスケ(注2)をくれ。
ようこ(以下Yo)：はい。
善：あぁ～，スイカソーダ2はうめぇ。このキュウリをかじりながらクリームソーダを飲むような青くささがたまらん。ところで長老，私，前から思っていたんですが，シフト命令とかローテイト命令って何に使うんです？　SHIFTキーと関係あるとか？
老：ないわけではないな。シフト(SHIFT)には“移動”とか“転換”とかいった意味があるようじゃが，ひらったくいえば“ずらす”といってしまって構わないようじゃ。ローテイト(ROTATE)は“回転”という意味だな。たとえばASCIIコードでAは41H，Bは42H，aは61H，bは62Hじゃから，小文字は大文字を20Hほどシフトさせているとも考えられるからのう。うむ，確かに初心者にはいったい何に使うのか戸惑うかもしれぬな。それでは，まずシフト命令から話そうか。
Yo：シフト命令っていうとぉ……，あ，SLAとかぁ，SRLとかぁ，SRAもあるっしぃ。
老：そ，そうじゃ(ようこくん，しばらく見ないうちに随分とミーハーな口をきくようになったな)。では，SLAが実際にやっていることを見てみようか。

A＝49Hが入っていたとして，これを2進表記すると，

A＝01001001(49H)

ここで，「SLA A」を実行すると，

A＝10010010(92H)となる。

つまり，各ビット列を左にずらし，一番右の空いたビットには0が入る，というわけだ。左からはみ出たビットはキャリフラグに取り込まれるというわけじゃな。
善：ふぅーん。それのどこに使い道があるというんです？
老：49Hは10進数にすると73，92Hは10進数にすると146。
Yo：わかった，2倍になっているのね。
老：そのとおり。これが10進数なら左シフトで10倍じゃ。
M：もう1度「SLA A」を実行するとどうなるんですか。
善：え，だから，

A＝00100100(24H)となってキャリフラグに1が入るんでしょ。
老：おぉ，西川くんにしてはよくできたな。キャリは8ビットからはみ出た桁だから，これを256(100H)と考えれば，

24H＋100H＝124H＝292

となる。
M：最初の49H(73)の4倍ですね。
老：そう。SLAをn回行えば2のn乗倍されるのじゃ。ちなみに，SLAはShift Left Arithmetic(算術的左シフト)の略じゃ。
Yo：右シフトにはSRAとSRLって，2つ命令があるみたいね。
老：ふむ。SRLの動作はまったくSLAの逆なんじゃ。こちらはShift Right Logical(論理的右シフト)の略だ。SRAはShift Right Arithmetic(算術的右シフト)で，少しばかり動作がSRLとは違う。

SRAはそれぞれのビット列を右にずらし，はみ出たビットはキャリに入る。ここまではSRAもSRLも同じじゃが，一番左のビットは変化しないというのがSRAの特徴だ。

だから，

A＝10010010(92H)

で，「SRA A」を実行すると，

A＝11001001(C9H)

となる。
善：ふぅん。名前からすると，SLAの反対ってSRAのようなのに，SRLがSLAの反対の働きをするのか。
M：覚え違いしそうですね。
老：しかし，名前のとおりSRAのほうが算術的じゃぞ。
善：なんで？
老：92Hを符号付きの数と考えれば－110，C9Hも符号付きの数と考えれば－55。
M：あぁ。なるほど。1バイトの符号付きの数では最上位ビット（7ビット）は符号ビットだから，SRAはそれを保存してシフ

トしているわけですね。
善：符号を変えない左シフトはないんですか。
老：うむ。6809（FM-7などのCPU）にもないようじゃな。少々動作が複雑だから実現できなかったのかもしれん。
Yo：あのう，ところで1バイトの符号付きの数ってなんですかぁ？
M：それはねようこちゃん，2の補数表現といってね，Xという数があったとして，全ビットを反転してから1を足したものを－Xと考えるといろいろな面で都合がいいんだ。コンピュータの世界ではこれを負の数として扱うことが多いんだよ。
善：そう。この2の補数表現では最上位ビットが符号ビットとなっていてこのビットが1だと負，0だと正というわけ。
老：たとえば－3を考えてみようかのう。

$03_H=00000011_B$

全ビット反転して，

$FC_H=11111100_B$

1を足すと，

$FD_H=11111101_B$

これをマイナス3と考えるわけじゃ。試しに10－3を計算してみようか。

$10=00001010_B$
$-3=11111101_B$
100000111_B

ここで答えは107_Hとなるわけだが，今は8ビットで数を考えていたわけじゃから，はみ出た部分を無視する。そういうわけで答えは7と。

回る回るローテイト

善：シフトはだいたいわかったけどローテイトは？　ローストビーフの親戚とか。
老：（無視して）ローテイト命令は単なるローテイトとローテイトサーキュラの2通りがある。
善：あの十字架とにんにくが嫌いという。
老：（さらに無視して）つまり，
・左ローテイト
RLC　(Rotate Left Circular)
RL　(Rotate Left)
・右ローテイト
RRC　(Rotate Right Circular)
RR　(Rotate Right)
があるのじゃ。
善：あの昼は棺桶で眠り，夜には生き血を求めてさまようという。（酔ってる……）
老：（どんどん無視して）ここでは左ローテイトについて話すことにするぞ。右についてはその逆だと考えてほしい。まずはRLCじゃ。
善：あ，それで回転寿司に入ると言われるのか「あー，いらっしゃー（RLC）」って。（回転寿司は左回りです，念のため。それにしても強引なダジャレ）
Yo：あははっ，すっご～くヘンな人～。
M：ダメだようようこちゃん，相手にしちゃ。ヘンが移っちゃうから。
老：RLCはレジスタの内容を左向きに回転させる命令なのじゃ。
善：と，いうとEが山になったり，Hが工になったりRがЯになったりするとか？
老：おいこら西川くん！　私は君がローテイト，シフト命令を教えてほしいというから……。（長老，コワイ目でにらむ）
善：……。
M：反省しているみたいです。
善：……ぐーっ，ぴーっ，……すーっ。
Yo：寝てますね……。
老：ほっときたまえ。まあ睡眠学習になるじゃろ。えー，気を取り直してと。RLCはまず，SLAのときと同じようにまず左にビットをずらす。はみ出たビットはキャリに入る。
M：ここまではSLAとまったく同じですね。
老：そして，一番右の空いたビットには先ほどはみ出たビットが入る。
たとえば，

$A=10101010_H(AA_H)$

を「RLC A」を実行後は（注3），

$A=01010101_H(55_H)$

となり，キャリには1が入っておる。これでわしがさっき回転と言った意味がわかったじゃろ。
Yo：えぇー。それじゃあ，キャリと一番右のビットっていつも同じなんですね。
老：そういうことじゃ。ふぉっふぉっふぉ。
♪カランコローン
源光（以下光）：こんばんは。
M：いらっしゃい。
光：おや，月の光がほのかに……と思ったらようこさんでしたか。
Yo：こんばんは，光君。
光：あ，今日はアサヒPF21ね。ところでここに寝ているビッグバンのような頭をしたオブジェはなんなのです？
老：あー，わしの弟子の西川じゃよ。
光：ふっ，長老も地に落ちましたねぇ，アヒルまで弟子にするんですか？
老：む。これにも一応知恵はあると思うのじゃが……まぁよい，いまローテイト命令の講釈をしておるからおぬしも聴くがよい。えー，今度はRLじゃがRLCのときと同じように全ビットが左にずれて……。
光：一番右の空いたビットにはキャリが入り，左ずれによってはみ出たビットはそのキャリに入ってくる。つまり，キャリを含めた9ビットのRLC命令とでもいいましょうか……。
Yo：わぁ，光君すごぉーい。
老：そのとおり。つまり，

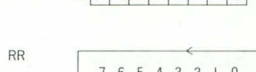
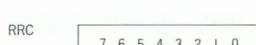
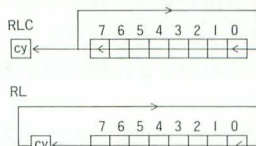
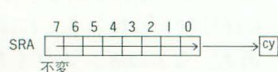
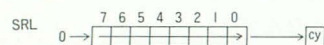
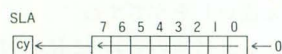
$A=10101010_H(AA_H)$

で，キャリに0が入っていれば「RL A」実行後で，

$A=01010100_H(54_H)$

となる。そしてキャリにはRL実行ではみ出たAの最上位ビットの1が入っとるというわけじゃ。
Yo：でもローテイト命令っていったい何に使うの？
光：うーん。たとえば，4回RLCやRRCを実行すれば上位4ビットと下位4ビットの交換できるし……。
老：RLやRRはキャリを取り込むことができる命令じゃから，計算の繰り上がり繰り下がりの処理などに使われるな。
M：あの，話は違うんですが，そういえばシフトやローテイトってオペランドにはA，

─MASTER'S MEMO─

シフト，ローテイト命令の働きを図で表すとこんなふうになる。

B, C, D……(HL), (IX+d) などの8ビットのものしか扱わないんですね。

老：うむ，でも今までの命令で同じような処理は可能じゃぞ。たとえば「SLA HL」などは，

```
    SLA    L
    RL     H
```

となるし，「SRL HL」は，

```
    SRL    H
    RR     L
```

となる。

Yo：すぅごぉーい。

光：もっとも「SLA HL」としたいならば「ADD HL, HL」というのがありますけどね。

Yo：光君すごぉーい。(こればっか)

老：なかなかやるな，おぬし。

M：あのう，レジスタの値を2のn乗倍するときはいいとして，たとえば6倍するときとかはどうするんです？

老：うむ。そう思うのもごもっとも。それは以下のようにするのじゃ。HL を6倍するとすれば，

```
    ADD    HL, HL    ;HL*2
    LD     D, H
    LD     E, L      ;DE=HL
    ADD    HL, HL    ;HL*4
    ADD    HL, DE    ;HL*6
```

とするのじゃ。

Yo：めんどくさーい。

光：だけどようこちゃん，掛ける数がわかりきっているときは掛け算ルーチンを使うより速いんだよ。

老：そうじゃな。もっともZ80に掛け算の命令があればそれに越したことはないんじゃがな。

Yo：ふぅーん。

人々のビットビット

老：あとはじゃ，ローテイト命令にはRLDやRRDがあるが，まあこれは2進化10進数の話が出てきたときに取り上げるとして，ほかにビットをいじくる命令にSET, RES, BITなどがある。

M：これは名前からもなんとなく想像できますね。「SET 1, A」としたとするとAレジスタの第1ビットをセット（1にする）して，「RES 5, A」としたとすればAレジスタの第5ビットをリセットする（0にする）んでしょう？

光：そう。BITは指定したビットを反転してゼロフラグに取り込む命令。「BIT 3, A」はAレジスタの第3ビットを反転してゼロフラグに取り込む。

Yo：ビットの反転ってなぁに？

光：ビットが0なら1に，1なら0にすることだよ。

Yo：そうかぁ，じゃ，今日説明してもらった符号付きの数なんかの正負を判断するのにいいわね。たとえば，

```
    BIT    7,A
    JR     NZ, MINUS
```

とかすればAの符号ビットを検査してAの正負を確かめたりできるわ。

光：そうだね，その組み合わせで疑似的に，

```
    JR     M, MINUS
```

として使えるね。

善：ふぁぁーあ。(大きなアクビをして)あれぇ，長老，ずいぶん若返っちゃってオマケに別人かと思うほどいい男になりましたねぇ。RLCはどうなりました？　むにゃむにゃ。

光：おやおやアヒル君，酔いは醒めましたか？

善：えっ？　どこどこ，どこにアヒルがいるの？

光：いえいえ。

善：あっもうこんな時間だ，これ今日の飲み代ね。(といってプログラムをひとつ置く)それじゃ，バイバイ。

♪カランコローン，……バタン

老：ふむ，西川君どこまで作ったのかな？アダルトっていってたっけ。

光：ゼンジソフト第1回作品「グラフィック画面消去プログラム」だって。……あれ？画面が消えるだけだ。

M：ということは，こっから先はまだ……。

光：この先は「読者のご想像におまかせします」かぁ。

Yo：ひょっとしてカレ，すごい想像力の持ち主なんじゃない？

光：うーん，こわいやつ……。

—つづく—

注1）かの有名な（かどうかは知らないが）スイカソーダのパート2である。キリンが出しているのが驚きだ。

注2）「つづく」の3文字を残し，いつの間にか消えてなくなった未完（？）の炭酸飲料。

注3） RLC A, RL A, RRC A, RR Aと同じ働きをする1バイト命令がある。それぞれRLCA, RLA, RRCA, RRAがそれで，処理速度も半分である。これはZ80の8080と互換性を持たせるために設けられた命令である。

リスト1　X1用グラフィック画面消去プログラム(このリストはZ80の一般的なアセンブラでアセンブルできます)

```
0000                1 ;*********************************
0000                2  ;     ゼンジソフト第1回作品
0000                3  ;グラフィック画面消去プログラム
0000                4 ;*********************************
E000                5         ORG     0E000H
E000                6
E000 0E 77          7         LD      C,01110111B       ;MASK PATTERN
E002 06 04          8         LD      B,4               ;LOOP COUNTER
E004                9                                   ;なぜ4かというとﾏｽｸﾊﾟﾀｰﾝが以下のように
E004               10                                   ;変化するからである。
E004               11         ;01110111 → 00110011 → 00010001 → 00000000
E004               12         ;                         (これで完全に消える)
E004               13 MAIN_LOOP:
E004 CB 09         14         RRC     C                 ;ﾏｽｸﾊﾟﾀｰﾝを右ｼﾌﾄ
E006 C5            15         PUSH    BC
E007 16 64         16         LD      D,100             ;D=100  DにY座標
E009 06 00         17         LD      B,0               ;B=0    Bはﾙｰﾌﾟｶｳﾝﾀｰで256が入っている。
E00B               18 CLR_LP0:
E00B 79            19         LD      A,C
E00C 0F            20         RRCA                      ;ﾏｽｸﾊﾟﾀｰﾝを右ｼﾌﾄ
E00D 0F            21         RRCA                      ;ﾏｽｸﾊﾟﾀｰﾝを右ｼﾌﾄ
E00E 4F            22         LD      C,A
E00F 5F            23         LD      E,A               ;E=ﾏｽｸﾊﾟﾀｰﾝ
E010 7A            24         LD      A,D
E011 C6 75         25         ADD     A,117             ;Y座標に117を足す.ここの値を変えると画面の
E013 57            26         LD      D,A               ;消え方も変わる。
E014 FE C8         27         CP      200               ;Y座標が200を越えてしまったか？
E016 30 42         28         JR      NC,OVER_200
E018 C5            29         PUSH    BC                ;ﾙｰﾌﾟｶｳﾝﾀｰとﾏｽｸﾊﾟﾀｰﾝをｾｰﾌﾞ
E019 E6 07         30         AND     7                 ;Y座標を8で割った余りを求める。
E01B 87            31         ADD     A,A               ;x2
E01C 87            32         ADD     A,A               ;x4
E01D 87            33         ADD     A,A               ;x8     これを8倍にして、
E01E 67            34         LD      H,A
E01F 2E 00         35         LD      L,0               ;HL=A
E021 E5            36         PUSH    HL                ;       とっておく。
E022 6A            37         LD      L,D
E023 CB 3D         38         SRL     L
E025 CB 3D         39         SRL     L
E027 CB 3D         40         SRL     L                 ;Y座標を8で割る。(ｷｬﾗｸﾀｰ座標に変換される。)
E029 26 00         41         LD      H,0
E02B 29            42         ADD     HL,HL             ;x2
E02C 29            43         ADD     HL,HL             ;x4
E02D 29            44         ADD     HL,HL             ;x8
E02E 29            45         ADD     HL,HL             ;x16
E02F 44            46         LD      B,H
E030 4D            47         LD      C,L               ;BC←HLx16
E031 29            48         ADD     HL,HL             ;x32
E032 29            49         ADD     HL,HL             ;x64
E033 09            50         ADD     HL,BC             ;HL←HLx64+HLx16  ｷｬﾗｸﾀｰ座標を80倍して、
E034 C1            51         POP     BC                ;先程とっておいたものを
E035 09            52         ADD     HL,BC             ;これと足して、
E036 01 00 40      53         LD      BC,4000H
E039 09            54         ADD     HL,BC             ;4000Hをさらに加えれば
E03A 44            55         LD      B,H
E03B 4D            56         LD      C,L               ;BCは完全な(青の)G-RAMのｱﾄﾞﾚｽとなる。
E03C               57
E03C 26 50         58         LD      H,80              ;横は640ﾄﾞｯﾄ÷8=80であるから
E03E               59                                   ;ｶｳﾝﾀｰとして80を使う
E03E               60 CLR_LP1:
E03E ED 78         61         IN      A,(C)             ;G-RAMからﾃﾞｰﾀを読む。
E040 A3            62         AND     E                 ;ﾏｽｸﾊﾟﾀｰﾝとANDをとって
E041 ED 79         63         OUT     (C),A             ;書きこむ。
E043               64
E043 78            65         LD      A,B
E044 C6 40         66         ADD     A,40H             ;BC←BC+4000H
E046 47            67         LD      B,A               ; これでBCは赤のﾌﾟﾚｰﾝのｱﾄﾞﾚｽとなった。
E047 ED 78         68         IN      A,(C)
E049 A3            69         AND     E
E04A ED 79         70         OUT     (C),A             ;青のときと同様の操作をする。
E04C               71
E04C CB F0         72         SET     6,B               ;BC←BC+4000H
E04E               73                                   ; これでBCは緑のﾌﾟﾚｰﾝのｱﾄﾞﾚｽとなった。
E04E               74                                   ;(実際に4000Hを足よりこちらの方が
E04E               75                                   ;速いためこうしている。)
E04E ED 78         76         IN      A,(C)
E050 A3            77         AND     E
E051 ED 79         78         OUT     (C),A             ;青のときと同様の操作をする。
E053               79
E053 CB B8         80         RES     7,B               ;青のﾌﾟﾚｰﾝのｱﾄﾞﾚｽに戻す。(BC←BC-8000H)
E055 03            81         INC     BC
E056 25            82         DEC     H
E057 20 E5         83         JR      NZ,CLR_LP1        ;80回分LOOPしたか？
E059 C1            84         POP     BC
E05A               85 OVER_200:
E05A 10 AF         86         DJNZ    CLR_LP0           ;256回分LOOPしたか？
E05C C1            87         POP BC
E05D 10 A5         88         DJNZ    MAIN_LOOP         ;4度LOOPしたか？
E05F C9            89         RET
```

X1/X1turbo用
バッハのイタリア協奏曲

Hanai Akiyoshi
花井 章能

X1/X1turbo用
代々木ゼミナール校歌

Itou Keiichi
伊藤 圭一

X68000用 ©KONAMI
サンダークロスよりFirst Attack

Tachikawa Masayuki
立川 正之

X68000用 ©日本ファルコム
ソーサリアン
「呪われたクイーンマリー号」より
船内のテーマ

Nishikawa Zenji
西川 善司

さて，先月はページの都合によりお休みしてしまい申し訳ありませんでした。今回は気合の入った4本，ゲームミュージックはもちろんのことバッハのイタリア協奏曲に代々木ゼミナール校歌までユニークな作品が揃っています。残暑たけなわ(?)の季節ですが，はりきって聴いてみてください。

baroque for MusicBASIC

今月の1曲目はMusicBASIC用イタリア協奏曲。1734年に作曲された，バッハの作品の中でもポピュラーな曲のひとつです。

プログラムは第1楽章から第3楽章の各章がリスト1からリスト3にそれぞれ収められています。3楽章を続けて聴くには，リスト1の110行にある ' を取ってRUNし，画面にPLAY"X"と出たらリターンキーを押して演奏を始めてください。

このプログラムでは各パートデータを一括してバッファに送り込んでいるのでRUNしてから曲が始まるまでに多少時間がかかります。変数領域を確保してまとめてバッファに送り込むという，テンポをずらさないためのこのテクニックはPLAY"Z"のないX1ユーザーとしてはかなり有効だと思います。一時期はやったA$(1)="……", A$(2)="……"のやり方に似ていますが，はるかにスマートですし，入力したところまできっちり聞けるというのはとってもオイシいというわけで，X1ユーザー以外の人もぜひプログラムをチェックしましょう。簡単な仕掛けで効果もバッチリなのですが，メモリの消費量は大きいのでうまく使ってください。

また，このプログラムでもやはりLABEL"!"をプログラムの頭のほうに持っていくだけで演奏開始がかなり早くなります。

さあて，お次は

なんと代々木ゼミナールの校歌です。学生手帳の4，5ページに載っている楽譜をMML化して送ってきてくれました（リスト4）。うーん，Oh!Xの読者ってなんてレパートリーが広いんだろう。

このプログラムは祝版，MIDI，MusicBASICという，これまでOh!Xで発表されてきたMMLならばすべてで演奏可能です。音色DATAとして，VIPに入っている"CHOICED VOICE.VTD"をB000番地にロードしてからRUNしてください。

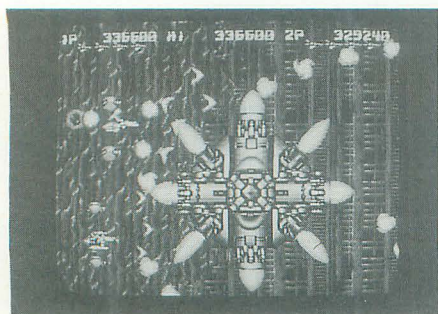
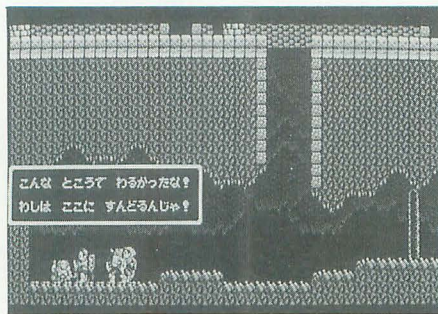
代ゼミといえば日本3大予備校のひとつですが，その校歌を載せてしまうあたりはさすがに受験情報誌Oh!Xならではといった感じがありますね。クラシックからゲームミュージック，予備校校歌までと今月の選曲を見てもらえば一目瞭然ですが，Oh!Xではスタッフに受ければなんでもありなのです。あなたもこれなら受ける！　と思った曲はビシバシMML化しましょう。軍艦マーチだって花の子ルンルン（知らないかな？）だってアレンジしちゃったり，また"演歌版アウトラン・こぶしをきかせてつっぱしれ！"とか，オリジナルソング"消費税にエンズイギリ"など，いままでに載っていなかったようなジャンルの曲も大歓迎ですよ。　(S.K.)

サンダークロス発進！

さて，X68000用にコナミのサンダークロスよりFirst Attackです。この曲は"びよよ～ん"ポルタメントや，"うにょうにょ"ビブラートがとっても気持ちよくて，「うん，こいつはいいぞ，なかなか好きだぞ」ということになり，プログラムしたのです。

得意分野のゲームミュージックなのでいくらか楽でしたが，さすがに冒頭に出てくる上向きのなが～いポルタメントを再現するのは疲れました。初めは「音符を粗くしてごまかしてしまおう」とか「サブルーチンを組んでポルタメントデータを作らせようかな」とか考えていたものの，そこはさすがに(?)うさんくさい人間の私，音が粗くなるのは嫌だし，サブルーチン組むのも大変だし，「あ～めんどくさ，Yコマンド並べよ」ということになりました。

リスト5を見てきっと気が遠くなると思いますが，一応，行の使い回しができるようにしてあるので気長にやってください。

このプログラムはOPMAに対応しています。おかげでサンプリングドラムもバシバシ，FM音源のみのアーケード版よりもカッコよくなってしまいました。もちろん，FM音源でドラムを出力したうえでサンプリングを重ねていますから，このままでも

サンダークロス

ソーサリアン

十分聴きごたえがあります。ただ，気になる人は9000行あたりの文字変数のなかの「Y2, ～」をすぱすぱと消してください。

今回はサンプリングパートはまったく凝りませんでしたが，ヤル気とOPMAのある人はシンバルをがしゅがしゅ鳴らしたりクラップをぱぱんがぱんと入れたりして，楽しんでみてください。　（立川）

お馴染みソーサリアン

やっぱりソーサリアンはいいなあ，というわけで，シナリオ「クイーンマリー号」の中から船内のテーマを同じくX68000のOPMA用にアレンジして作ってみました。プログラムはリスト6です。

X1用ではPSGで鳴っていた裏メロをFMシンセブラスで鳴らして，リズムはもちろんサンプリング。いい雰囲気が出てるでしょう。FM音源のほうでは，皆さんにもすっかりお馴染みとなったアルゴリズム4のピアノ(@72)やエレクトリックオルガン(@76)などにハードLFOをかけたものに，さらにKF（キーフラクション）をずらしたものと重ねて，簡単なエコー効果を出してみました(リスト中のfunc vset2( )を参照してください)。この曲を，ソーサリアンを知らない友人に聴かせたら，彼いわく「侍ジャイアンツみたいだね。ほら，おれはさむーらいー，ってね」といってましたっけ。

ところで，7月号に掲載されたボスコニアンのBlast Powerでは，茨城県の田中靖浩さんのMMLを一部参考にさせていただきました。記載がもれてしまったことをお詫びいたします。　（西川）

MusicBASICは，本誌1988年12月号で発表されたX1/X1turbo用MMLです。OPMAは，同じく1989年4月号掲載の，FM/PCM音源の同期演奏をMMLで行うX68000用ミュージックドライバです。

リスト1　イタリア キョウソウキョク.1

```
10 REM ******************************
20 REM *                            *
30 REM *   ITALIENISCHES            *
40 REM *                  KONZERT   *
50 REM *                            *
60 REM *               by J.S.BACH  *
70 REM *                            *
80 REM ******************************
100 TEMPO 0
110 'F=1
120 IF F=1 THEN PLAY "X"
130 DIM A$(50),B$(50),C$(50),D$(50)
140 K=0
150 MEM$(&HB190,36)=HEXCHR$("FA413177653226392D009CDB9CDA0409048
203C10300170206F50080800000BD85050200")
160 A$="I01Q8":B$=A$:C$=A$:D$=A$
170 "!"
180 I=0
190 A$="V15O3L8F>>F<AB->C4D4CF4E-16D16DCR4"
200 B$="V15O3L4C>F8G8AB-AB-B-8A8R"
210 C$="V15O2L4ARR2>>F2FR"
220 D$="V15O2L8F4R4>>RFD<B-RFD<B->F<F>R4"
230 IF I=1 THEN MID$(A$,5,5)="4L8F "
240 "!"
250 A$="<C>>C<EFG4A4G>C4<B-16A16AGL16RC<B-A"
260 B$="<G>>C8D8EFEFF8E8R4"
270 C$="<ERR2>>C2CR"
280 D$="C4R4>>RC<AFRC<AF>C<CR4"
290 "!"
300 A$="B-8>G8.<B-AGAGFGAB->CDE-8>C8.<E-DCDC<B->CDEFG"
310 B$="<L8C4C4.<A>CFF4F4.DFB-"
320 C$="<<GGFR>CC<B-R"
330 D$="L4ECR2AFR2"
340 "!"
350 A$="A-G32F32G8.F32E32F8.E32D32E8.GEC"+STRING$(4,"<B-32R32>GE
C")
360 B$="B4G4D4C4RC16D16EGEC<C4"
370 C$=">FD<BGR1"
380 D$=">D<BGCR1"
390 "!"
400 A$="<A32R32>AFD-C32R32AFD-C32R32AFD-C32R32AFE-"+STRING$(4,"D
32R32BA-F")
410 B$=">R16C16D16E16FAFC<C4R>D16E16FD16E16FD<GB"
420 C$="R1R1"
430 D$="R1R1"
440 "!"
450 A$="E>C<GECEDFEGEC<GB-A>C<B->D<B-GEGFAGB-GECB-AG"
460 B$=">C<C16D16EG>C<C16D16EFGC16E16GAB-C16D16EC"
470 "!"
480 A$="AFGAB->CDEFEDC>C8<F8F4E4&EF8.R4RCF4E8F4R8C8"
490 B$="L16FAGFEDC<B-A>CDEFGAB->>C4C4.<B8.>DC<B-A8.AL8G>CC<V12AA
A"
500 C$="R2>>R4.C8<AGFRL8R4.B-AV12FFF"
510 D$="R2>R4.A8C<CDL8REF<AB->C<FR4."
520 IF I=1 THEN 1970
530 "!"
540 A$="DCD8.FE-DCEF8.EDC<B-AB->DG8<B-8>C32<B-32>C32<B-32A8R8A8"
550 B$="B-B-B-B-B-B-AAGGGGGFRF"
560 C$="FFFFFFFFFFEEF<F16G16AF>"
570 D$="R1R1"
580 "!"
590 A$="GB-B-AA>FFEEDDCC<B-B-AGB->C32<B-32AA>FG32F32EL32FED16EDC
16DC<B-16>C<B-A16"
600 B$="FEFAGFEFFEFAGFEG"
610 C$=STRING$(16,"C")
620 "!"
630 A$="B-AL16GA8.>C<B-AGB>C8.<B-AGFEFA>D8<F8G32F32G32F32E8R>CDE
"
640 B$="FFFFFFEEDDDDDCR4"
650 C$="CCCCCCCCCC<BB>C<C16D16EC"
660 "!"
670 A$="<AEFA>DDEF<BF+GB>EEFGC<G+A>CF+F+GA"
680 B$="L16<RGA>CF8R8<RAB>DG8R8<RB>CEA8C8"
690 C$="F4RDG4REA4RA"
700 D$="R1R2"
710 "!"
720 A$="A32G32F+G8.<GB>DFEF8.<GB>DEDE8.<G>CD"
730 B$="L8D<B>DFD<B>DFL16GFEDC8R8"
740 C$="BGB>D<BGB>DL16C4&C<BAG"
750 "!"
760 A$="E-C<B>CF+CAC<BFEFBF>D<FEG>CGFEDCC4R4"
770 B$="<R2R8D8G4.C4<B8>C>V15C<BAGFED"
780 C$="F+EDC<BAGF+G4L8RB>C<EFGV15R2"
790 D$="R1R1"
800 "!"
810 A$="R8C8<E8F8G4A4GAB-4AGA4R4"
820 B$=">R4C8D8E4F4E2RFEDCR8."
830 C$="R2>>RC<AFC2&CRL16RB-AG"
840 D$=">L16CC<BAGFEDC8R2A8E8C8F8R4."
850 "!"
860 A$=">R8F8<A8B-8>C4D4CDE-4DCD4R4"
870 B$="R4F8G8A4B-4A2RB-AGFR8."
880 C$="R2>R8F8D8<B-8F2&F8R8.>E-DC"
890 D$=">FFEDC<B-AGF8R2>C8<A8F8B-8R4."
900 "!"
910 A$="RDC+DB-8D8.C+<B>C+A8C8.<BAB>G8<B-8>C32<B-32>C32<B-32A8R4
920 B$="<B-8>G8.<B-AGA8>F8.<AGFG8>E8.DC+D<G>DC+E<AGFE"
930 C$="R1R1"
940 D$="R1R1"
950 "!"
960 A$=">RD<B-GA>D<GEF>D<GEF>D<EC+DV13>D<B-32A32GA>D<G32F32EF>D<
G32F32EF>D<E32D32C+"
970 B$="L8FGFEDB-AGFV13GFEDB-AG"
980 "!"
990 A$="L8RV15>AB-ARGAGRFGFRFGF&"
1000 B$=">DV15>FGF<E>EFE<F>DED<G>DED"
1010 C$="<F>C+DEDCD<F32R32G32R32>C+DEDCD<G32R32A32R32>C+DEDCD<A3
2R32B-32R32>C+DEDCDF32R32"
1020 "!"
1030 A$="F16L32EDE8.DC+D8.C+<B>C+8.L16EC+<A"
1040 B$="<G+4E4<B4A4"
1050 C$="D4<B4G+4E4"
1060 D$=">B4G+4E4<A4"
1070 "!"
1080 A$=STRING$(4,"G32R32>EC+<A")+"F32R32>FD<B-A32R32>FD<B-A32R3
2>FD<B-A32R32>FDC"
1090 B$="RA16B16>C+EC+<A<A4>R16A16B16>C+16DFD<A<A4"
1100 C$="R1R1"
1110 D$="R1R1"
1120 "!"
1130 A$=STRING$(4,"<B32R32>G+FD")+"C+AEC+<A>C+<B>DC+EC+<AEGFA"
1140 B$=">RB16>C+16D<B16>C+16D<BEG+A<A16B16>C+EA<A16B16>C+D"
1150 "!"
1160 A$="GB-GEC+EDFEGEC+<A>GFEFDEFGAB>C+DC+<BA>A8D8"
1170 B$="E<A16>C+16EFG<A16B16>C+<A>L16DFEDC+<BAGFAB>C+DEFG"
1180 C$="R1R2.R8>A8"
1190 "!"
1200 A$="F4E4&ED8.R4R<A>D4C+8D4R8<A8"
1210 B$=">>D4C+4&C+D<G+8.B-AGF8.FE8A8A4R4"
1220 C$="L4AGFR2R8G8FR"
1230 D$="L8>A4<A4B-4R>C+D<FGADV12L16>DEFGAB"
1240 "!"
1250 A$="B-32A32G+A8B-32A32G+A8.B>C+DE8<A8B-32A32G+A8B-32A32G+A8
.>C+DEF8<A8"
1260 B$="V12L8ERFRER4.FRERL16FGFEDEFD"
1270 C$="V12L8C+<A>D<A>C+<L16AB>C+DEC+L8D<A>C+<A>DR4."
1280 D$="R1R1"
1290 "!"
1300 A$="B-32A32G+A8B-32A32G+A8.>C+EFG8<A8.>GFE32D32B-8<G8.>FED3
2C32A8<F8&"
1310 B$="G8R8F8R8EDC+<BAGFED>DEF<G>FED<C>CDE<F>E-DC"
1320 C$="E<A>D<A>C+R4.R1"
1330 "!"
1340 A$="F>E-DC32<B-32>GDC<B-32A32>FC<B-A32G32>E-<B-AG32F32>D<AB
->D<E->DC<B-AGA>C<F>E-DC"
1350 B$="<<B->B->CDL8<E-ADGCF<B->DGE-FC<AF"
1360 C$="R1R1"
1370 "!"
1380 A$="L8DB-DE-F4G4FB-4A-16G16GFRF"
1390 B$="V15>>R4B->CD4E-4D4E-4E-DRD"
1400 C$="V15R1B-2.G4&"
1410 D$="L8V12<B-<B-R4>>>V15RB-GE-R<B-GE-L16B->CDC<B-AB-G"
1420 "!"
1430 A$="FE-RE-E-DRDDCRCC<B->B-4.L16AGAF+GA"
1440 B$="DCRCC<B-RB-B-ARAAGL16>RDGFE-4R4"
1450 C$="G4F2E-2D2R2."
1460 D$="CG>C<B-AGAF<B->FB-AGFGE-<A>E-AGF+EF+D<G>DGAGFE-DCG>CDC<
B-AG"
1470 "!"
1480 A$="L32CD"+STRING$(25,"E-D")+"<G8B-8>D8"
1490 B$="<<F+>DC+D<DF+EGF+ADF+GB-DGA>C<AF+D>C<B-AB->D<B-GDFED"
1500 C$="R1R1"
```

```
1510 D$="R1R1"
1520 "!"
1530 A$="<B>C"+STRING$(25,"DC")+"<F8A8>C8"
1540 B$="E>C<B>C<CEDFEGCEFACFGB-GECB-AGA>C<AFCE-DC"
1550 "!"
1560 A$="L16F8<F8B-2A2&AAGFEGEDC4"
1570 B$="DFD<B-GB-AG>CEC<AFAGFL8B->CD<B->CDEC"
1580 C$="L16R4R8D8EGECR8C8DFDC<B-2&B-B-AG
1590 "!"
1600 A$="<A>CFEDC<B-AB->EGFEDC<B->CFAGFEDCDFB-AGFEDECEGB->CEG"
1610 B$="<CAB->D<CB->CE<C>CDF<C>DEG<C4R4"
1620 C$="R1R1R2"
1630 D$="R1R1R2"
1640 "!"
1650 A$="A8D8EDC+D8GD<B>C<B>CEF8<B-8>C<B-AB-8>E-<B-GAGA>CD8<G8AG
F+G8>C<GEFEFA"
1660 B$=">>V12RAAAGGGGRFFFE-E-E-E-RDDDCCCC"
1670 C$=">V12L8RFFFEEEERDDDCCCC<RB-B-B-AAAA"
1680 D$="R1R1R1"
1690 "!"
1700 A$=STRING$(2,"L32AB-"+STRING$(8,">C<B-")+"L16>GEC<B->GEC<")
1710 B$="L16RDEF<G>E<F>D<EG>C8R4<RGFE>C<E>D<D>E8<C8R4"
1720 C$="R1R1"
1730 D$="R1R1"
1740 "!"
1750 A$="A8>F8<A8B-8>C4D4.CDE-8DCD4R4"
1760 B$="V15>L8R4FGA4B-4.A4.B-4R4"
1770 C$="V15L16>RFEDC<B-AGFFE-DC<B-AGF>FC<AF>FE-F<B->DFAB->DFA"
1780 "!"
1790 A$="R8G8<B8>C8D4E4.DEF8EDE4R8C8"
1800 B$="R4GAB4>C4.<B4.>C4R4"
1810 C$="GGFEDC<BAGGFEDC<BAG>GD<BG>GFGCV12GAB>C<G>CD"
1820 "!"
1830 A$="D32C32<B>C8D32C32<B>C8.DEFG8C8D32C32<B>C8D32C32<B>C8.EF
GA8C8"
1840 B$="V12<GRARGR4.ARGRL16AB-AGFGAF"
1850 C$="L8ECFCEL16CDEFGEL8FCECFR4."
1860 "!"
1870 A$="D32C32<B>C8D32C32<B>C8.EGAB-8C8.B-AG32F32>C8<E-8.D32C32
DC32<B-32>G8<B-8.A32G32AG32F32>D8<F8G32F32G32F32E8R8>B-8"
1880 B$="B-8R8A8R8GFEDC<B-AGFGFE-DC<B-AB->FB-8.GFEFA>C8.<BAB>C<G
>CDECDE"
1890 C$="GCFCER4.R1R1"
1900 D$="R1R1R1"
1910 "!"
1920 A$="AB->C<FAGFEFCDF<B-AB->DF<B->CF<AGA>CF<AB>F<GFG>FEFGFEDC
<B-AB->C<FAGFE"
1930 B$="L8F<AB->CD<B-F>DC<AF>C<B>D<BG>C<CDEF<AB->C"
1940 C$="R1R1R1"
1950 "!"
1960 I=1:GOTO 190
1970 A$=LEFT$(A$,20)+"T95F4E4&EF8.R4T90RCF4T80E8F1"
1980 B$=LEFT$(B$,50)+"1"
1990 C$=LEFT$(C$,22)+"1"
2000 D$=LEFT$(D$,25)+"1"
2010 "!"
2020 PLAY "T100P1";    :FOR I=0 TO K-1:PLAY A$(I);:NEXT
2030 PLAY ":P2K10R@2";:FOR I=0 TO K-1:PLAY A$(I);:NEXT
2040 PLAY ":P1";       :FOR I=0 TO K-1:PLAY B$(I);:NEXT
2050 PLAY ":P2K10R@2";:FOR I=0 TO K-1:PLAY B$(I);:NEXT
2060 PLAY ":P1";       :FOR I=0 TO K-1:PLAY C$(I);:NEXT
2070 PLAY ":P2K10R@2";:FOR I=0 TO K-1:PLAY C$(I);:NEXT
2080 PLAY ":P1";       :FOR I=0 TO K-1:PLAY D$(I);:NEXT
2090 PLAY ":P2K10R@2";:FOR I=0 TO K-1:PLAY D$(I);:NEXT
2100 PLAY ""
2110 IF F=1 THEN PLAY "R1":CHAIN "ｲﾀﾘｱ ｷｮｳｿｳｷｮｸ.2"
2120 END
2130 LABEL "!"
2140 A$(K)=A$:B$(K)=B$:C$(K)=C$:D$(K)=D$
2150 K=K+1
2160 RETURN
```

リスト2 イタリア キョウソウキョク.2

```
100 IF F=0 THEN TEMPO 0
110 IF F=0 THEN DIM A$(20),B$(20),C$(20)
120 K=0
130 MEM$(&HB190,36)=HEXCHR$("FA413177653226392D009CDB9CDA0409048
203C10300170206F50080800000BD85050200")
140 A$="I01Q8":B$=A$:C$=A$:D$=A$
150 "!"
160 A$="R1R1R1R1R2"
170 B$="V12L4O4FRRFGAB-RRAGFERRGFE"
180 C$="V12L4O4D<DD>DEFG<DD>FEDC+<DD>EDC+"
190 "!"
200 A$="V14L8O5A16G16A2GFEDC+DF8.E16DE4.GAB-4C+DE<A16G16FG4.AB->
C+4D4E&"
210 "!"
220 A$="E<GG32F32G32F32EFB-B-32A32B-32A32GAL16B>C+DEFGFEDEF8D8C8
<B-1AGFGE8E8A4.&A2&"
230 B$="FRREDC+DRRFEDERREDC+"
240 C$="D<DDGFED<DD>>DC<B->C<CC>C<B-A"
250 "!"
260 A$="A8B>C+DC+<BA>D8<B-8B-32A32B-32A32G8.AGF+G8.GG8>E4.&EDC+D
EDC+<BA8G8.B-AGFGFEFAC+8C+8D8R2>D4&"
270 B$="DRRDC+<B>C+RRC+DEFRRFGA"
280 C$="B-<B-B->B-AGA<AA>AB>C+D<DD>DEF"
290 "!"
300 A$="L8DCC32<B-32>C32<B-32AB-GB->DGDB-4&L16B->C<B-AB-GAB-F8E8
R2CDE8.<B->CDC<B-AGAFGAB->CDEFEDCAGA8&"
310 B$="B-RRAGFERREFGARRGFE-"
320 C$="G<GG>FEDC<CC>CDEF<FF>EDC"
330 "!"
340 A$="A8D8E-64D64E-64D64CD8D8G4.&G2&GB-AGFEDCC8F4.GFEFL32GFGFG
FGF&L16FEDEF8D8C<B-A8B-AB-8.>C<B-AB-8.B-"
350 B$="DRRDC<B->CRRC<B-AB-RRAGF"
360 C$="<B-<B-B->B-AGA<AA>AGFG<GG>FED"
370 "!"
380 A$="B-8>G4.&GFEFGFEDC8<B-AB-8.>G<B-8A8>A4&AGFGAGFEDC<B-AGFEF
F8E8.GFEB->C<B-AB-8>C4D4E8&"
390 B$="ERREFGARRFGAB-RRGAB-"
400 C$="C<CC>CDEF<CC>DEFG<CC>EFG"
410 "!"
420 A$="EGFEFCDEFEFGFGAGAB-AB->C<AB->CC8<F4E8.DEFEGFEDC<B-AB-8.>
GGFEGFEDFEDCEDC<B->DC<B-A>C<B-AGB-"
430 B$=">CRRC<B-AB-RRB-AGARRAGF"
440 C$="A<CC>AGFG<CC>GFEF<CC>FED"
450 "!"
460 A$="L8B-C4EGB->DC8.L16<B-AGA8>C8.<GFEF8>D8<AB-AGAGFGFG{AGAGA
G}4FGF2R1"
470 B$="GRRGAEF2.E8D8EB-ARRAB>C+"
480 C$="E<CC>EFCD<B->C1F<FF>GFE"
490 "!"
500 A$=">R2FEF1EDC+8D8<GAB-4B-8.>C<B-AB->C<B-AB-8>G4C+4E4G4B-4A8
&"
510 B$="DRRFGAB-RRAGFERRGFE"
520 C$="D<DD>DEFG<DD>FEDC+<DD>EDC+"
530 "!"
540 A$="AGFEF8D8.C+DEDC+<BA8.>D4F8.GFEF8B-4A4G4FEDC<B-AGB>DF8.E8
.DC<B>C8<G4>C4E8&"
550 B$="FRRFGAB-RR<B->CDERREFG"
560 C$="D<DD>DEFG<<GG>GAB->C<CC>CDE"
570 "!"
580 A$="EFE-DE-8>C8.<B-A8.GF8.E-DE-FE-DE-<FAGF>C<B-A>E-DC+D4.&D<
DC+DEFGAB->D<B-AB->D<B-AGF+G4.&GB-GFG>E<GF"
590 B$="ARR<AB->CDRRC<B-AGRRFED"
600 C$="F<<FF>FGAB-<B-B->AGFE<EE>DC+<B"
610 "!"
620 A$="G8B-8>C+8E8.DC+DEDC+<BA8GFG8.>E<G8F8>F4&FEDEFEDC<B-8AGA8
.>F<A8G8>G4&GFEFGFEDC+8<B-AB-8.>G"
630 B$="C+RRC+DEFRRDEFGRREFG"
640 C$="A<AA>AB>C+D<<AA>B>C+DE<<AA>>C+DE"
650 "!"
660 A$="<B-8A4>C+4D4EFGAB-AB-8.B-B-8C+4DEE8<G4AB-B->EFG8.<B-8A>D
EF8.<G+8.BAG+AB>C+DE<B-AG+AGF+G"
670 B$="ARRAGFGRRGFEFRRFED"
680 C$="F<<AA>>FEDE<<AA>>EDC+D<<AA>>DC+<B"
690 "!"
700 A$="GFG4.&GA>C+EGB-A8.GFEF8A8.EDC+D8B-8FGFEFEDEDE{FEFEFE}4DE
D4CDE-8.DC8.<B-A8.GF+EDC+D8&"
710 B$="ERREFC+D2.C+8<B8>C+GF+RRF+GA"
720 C$=">C+<<AA>>C+D<AB-GA1>D<DD>DEF+"
730 "!"
740 A$="D8EF+GDGAB-GB->CD<A>C+DEFGAB-DC+DD8<G8.AGFG8>C+DE8<G8FAB
>C+D8<B-8T60G+8A4DGF8&T50L32FGFEFEDEDC+DED16E16FEFEFEFET40D16E16
D1&D1"
750 B$="B-RRAGFGRREFAD1&D8C+8G2F4&F1"
760 C$="G<DD>FEDE<DD>C+DF<BB-AG+A2>D1&D1"
770 "!"
780 PLAY "T66P1";     :FOR I=0 TO K-1:PLAY A$(I);:NEXT
790 PLAY ":P2K10R@1";:FOR I=0 TO K-1:PLAY A$(I);:NEXT
800 PLAY ":P1";       :FOR I=0 TO K-1:PLAY B$(I);:NEXT
810 PLAY ":P2K10R@1";:FOR I=0 TO K-1:PLAY B$(I);:NEXT
820 PLAY ":P1";       :FOR I=0 TO K-1:PLAY C$(I);:NEXT
830 PLAY ":P2K10R@1";:FOR I=0 TO K-1:PLAY C$(I);:NEXT
840 PLAY ""
850 IF F=1 THEN CHAIN "ｲﾀﾘｱ ｷｮｳｿｳｷｮｸ.3"
860 END
870 LABEL "!"
880 A$(K)=A$:B$(K)=B$:C$(K)=C$
890 K=K+1
900 RETURN
```

リスト3 イタリア キョウソウキョク.3

```
100 IF F=0 THEN TEMPO 0
110 IF F=0 THEN DIM A$(50),B$(50),C$(50),D$(50)
120 K=0
130 MEM$(&HB190,36)=HEXCHR$("FA413177653226392D009CDB9CDA0409048
203C10300170206F50080800000BD85050200")
140 A$="I01Q8":B$=A$:C$=A$:D$=A$
150 "!"
160 I=0
170 A$="V15L8O5F4<F2GAB->CDEFGAB->C4C4C4<B-4A2G2"
180 B$="V15L4O5R1R1RAGGF2E2"
190 C$="V15L4O3A2>C2<FB-AGF8R8>>FECC2C2"
200 D$="V15L8O3F2E2D2C4<B-4AFGAB->CDEFGAB->CDEC"
210 IF I=0 THEN 260
220 A$="<A>C<F4.F"+MID$(A$,13)
230 B$="<RGRB->C2L4>"+MID$(B$,10)
240 C$="L4RA2."+MID$(C$,14)
250 D$="<F2.E4"+MID$(D$,12)
260 "!"
270 A$="A2RGECF2RE-C<A>D2RC<AFB-2RB-GE"
280 B$="L8RFEDC2RDC<B-A2RB-AGF2&FFEDE2"
290 C$="<F<FEA>D<DCFB-<B-A>D<G>G8A8GF"
300 D$="R1R1R1R1"
310 IF I=1 THEN C$=LEFT$(C$,20)+"G<G8A8GF>"
320 "!"
330 A$="CDEB-16R16AB->CF16R16EFGB-16R16AB->C<E-16R16DEF>C16R16<B
-AGFEFGEC2"
340 B$="L4<EGFAGCF<AB->D<GB->L8CDEFGAB-G"
350 C$="R1R1R1R1"
360 IF I=1 B$="L4<<EGFAG>E"+MID$(B$,10)
370 "!"
380 A$="F4<F2GAB->CDEFGAB->C4C4C4<B-4A2G2"
390 B$=">>L4R1R1RAGGF2E2"
400 C$="A2>C2<FB-AGF8R8>>FECC2C2<"
410 D$="<F2E2D2C4<B-4AFGAB->CDEFGAB->CDEC"
420 "!"
430 A$="L4B-2B-AG2GA":A$=A$+A$
440 B$="L8"+STRING$(4,"RFEFRF4F8")
450 C$="D<D>DC<B-<B->B->C":C$=C$+C$
```

▶MZ-80を持っている僕の友だちは,未だにYsIIのことを「ワイエス・ジューイチ」といっている。こんな僕の友だちに,誰かX1turboをプレゼントしてやってください。僕からもお願いします。
原　政徳 (16) 愛知県

```
460 D$="R1R1R1R1"
470 "!"
480 A$="B-AG2C<B-A2F>C<AG8F8F2R2"
490 B$="RF4FRFED<RG4GRGFEL4RFFEC2R2"
500 C$="DC<B-GECFC>RCC<B-A2R2"
510 D$="R1R1<<AGAB->C4<C4F4GAB->CDE"
520 IF I=1 THEN 2090
530 "!"
540 A$="V12L8RAB->C<B-AGF>E-CD4.<B-AB->E<B->F<B->G<B-AGA>C<B-AB-
AGA"
550 B$="<FA>C<AFB-8>C8D<FGB->ECFC<A>C"
560 C$="R1R1R1R1"
570 D$="R1R1R1R1"
580 "!"
590 A$="L4FV15A>C<AFB-8>C8D<FGB->ECF8E8F4.V12L8C<B->C"
600 B$="V12L8<FAB->C<B-AGF>E-CD4.<B-AB->E<B->F<B->G<B-AGA>C<B-AB
-AGA"
610 "!"
620 A$="AB-AGFEDC<B>G<BAG>DEFGAGFEDC<BA>F<AGF>CDE"
630 B$="V15F4A4>C4F4.EDC<BAGFE4G4>C4E4.DC<BAGFE"
640 "!"
650 A$="FGFEDC<BAG>F<GFG>F<A>F<B>F<BAB>FCFDFDCDFCF"
660 B$="L4DFA>C2<B8A8B>CD<G8F8GABB8A8B>C"
670 "!"
680 A$="<B>F<BAB>F<A>F<G>F<GFG>F<A>F<B>F<BAB>FCFDFDC<BAGF"
690 B$="D<G8F8GABB8A8B>CD<G8F8GAL8B>D<BAGFED"
700 "!"
710 A$="EGFEFEDCB-GA4.FEFBF>C<F>D<FEDEGFEFEDE"
720 B$="L4CEGECF8G8ACDFBG>C<GEG"
730 "!"
740 A$="L4CV15EGECF8G8ACDFBG>C<G>GF"
750 B$="L8CV12EFGFEDCB-GA4.FEFBF>C<F>D<FEDEGFEFEDE"
760 "!"
770 A$="L8E1&EGFEGFEFD1&DFEDFEDEC1&CEDCEDCD"
780 B$="V14>>RC<B-AB-AGAF4R2.RAGFAGFGD4R2.RGFEGFEFD4R2."
790 C$="!117CEGB-A>C<AF<B>DFAGBGE<A>CEGFAFD"
800 D$="R1R1R1R1R1R1"
810 "!"
820 A$="<B2.B4>C2R4E4F2.D4E2.F+4G1R1"
830 B$="R4D4G2R4G4>C2.<BAB2.AGA4>C4.EDCDC<BAGAB>CDEFD"
840 C$="L8<G16R16>GFEFEDE<G16R16>FEDEDCD<G16R16>EDCDC<B>C<G16R16
>DC<B>C<BABG16R16>C<BABAGAB>CDEFGAB"
850 "!"
860 A$="C4<C2DEFGAB>CDEFL4GGGFE2D2"
870 B$="V15R1R1R4E4D4D4C2<B2>"
880 C$="L4R1>RFEDC8R8>C<BGG2G2<"
890 D$=">C4R4<B4R4A2G4F4E<CDEFGAB>CDEFGABG"
900 "!"
910 A$="F2FED2DE":A$=A$+A$
920 B$=STRING$(4,"RC<B>CRC4C")
930 C$="A<A>AGF<F>FG":C$=C$+C$
940 D$="R1R1R1R1"
950 "!"
960 A$="FED2R8DD8R8D8C8<B8>CGED8C8C2R2"
970 B$="RC4CRC<BAR1R4>C4C4<B4G2R2"
980 C$="AGFD>GFE2.GGFE2R2"
990 D$="R1<B4G4>CDEFGFEFG4<G4C>V12CDEFGAB"
1000 "!"
1010 A$="V12L8<GECEGEB-EAFCFAF>C<FGECEGEB-EAFCFAFB-F>"
1020 B$="V12L4C<GEC>C<AFC>C<GECF8C8<A8>C8<F4R4"
1030 C$="R1R1R1R1"
1040 D$="R1R1R1R1"
1050 "!"
1060 A$="V13C<AFA>C<A>E-<A>D<B-FB->D<B->F<B->":A$=A$+A$
1070 B$="V13>FC<AF>FD<B-F>FC<AFB-8F8D8F8<B-4R4"
1080 "!"
1090 A$="V14GC+<B->C+GC+GDGE<A>EGEGD":A$=A$+A$
1100 B$="E4GFE4D4C+DC+<BAB>C+D":B$="V14L8>>"+B$+B$
1110 "!"
1120 A$="GC+<B->C+GC+GDGFGEFEDEFGA4<A4>C+4DV15DC<B-AGFE"
1130 B$="EFGAB-4D4C+<AB>C+DEFGAGFGA4<A4DV15>DEFGAB>C+"
1140 "!"
1150 A$="DEFGAB>C+DEFGEFEDEFDC+<B>C+DEC+D<AGFGAB-A"
1160 B$="D4<<D2EFGAB>C+DEFGARARARGRL16GFEFGFGFE2"
1170 "!"
1180 A$="B-1A1B-1A1"
1190 B$=">>L4RFEFRFGFRFEFRFGF"
1200 C$=">L4"+STRING$(4,"RDC+D")
1210 D$="V15GDC+DGDFDEDC+DEDFD":D$=D$+D$
1220 "!"
1230 A$=">E4D4C+DEFG2&GGFEF4A4F4EDD2R2R1R1"
1240 B$="<B-AGRR>E<A2.A2G4.L8AGFGFEFR>DC<B->C<B-A>C<B-2L16>C<B>C
<B>C<B>C<B"
1250 C$="C+R1R1DDC+R1<DG2F+L8RGFEFEDF"
1260 D$="GDFDEDC+<BA>E<G>E<F>EDC+D<AFDL4A<A>D>>D2C<B-2A2G1&"
1270 "!"
1280 A$="RGFEFEDFE4F4G2&G>C<B-AB-AGB-A4B-4>C2&C<FE-DE-DCE-D2C2"
1290 B$="L4>C2.<B>CDE2.F2ERGA2&A8R8R2.R1"
1300 C$="E2D2C<C<B-AB-AGB-A4>D4<G4>C4>>F1&FR<B-2A4.B-AGAGF+A"
1310 D$="G1R1R1<L8F>>FE-DE-DCE-D2C2<B-4>E-4<A4>D4&"
1320 "!"
1330 A$="<B-4<B-2>CDE-FGAB->CDE-F4F4F4E-4D2C2"
1340 B$="<GR2.RE-DC<B-8R8>>DCC<B-2A2&"
1350 C$="<GRR2.R1>R4B-4A4F4F2F2"
1360 D$="D<B-A-GA-GFA-G2F4E-4D<B->CDE-FGAB->CDE-FE-DF"
1370 "!"
1380 A$="RDC<B->C<B-A>C<B-4>G2F4.FEDEDC+E<A1&AAGFGFEGC+2R2"
1390 B$="AG2F+G2A2L8G1&GGFEFEDF<B-1&B-B-AGAGFE"
1400 C$="R1L4R>EC+DR1R1R1R1"
1410 D$="<B-2A2RGFEFEDFL4<B->DGEC+<A>D<FGB-EGAEC+<A>"
1420 "!"
1430 A$="R2C+DEFGAB>C+DEFGA4A4A4G4F2E2"
1440 B$="FGABR2R1>>L4RFEED2C+2"
1450 C$="R1R1RDC+<AA2R2"
1460 D$="DR2.L8RFGAB>C+DEFDEFGAB>C+DEFGAFGA"
1470 "!"
1480 A$="V12FGFEDC<BAG>F<GFGB>CDEFEDC<BAGF>E<FEFAB>CDEDC<BAG+F+"
1490 B$="<DAFD<B>D<BG>CGEC<A>C<AFB>FD<B"
1500 C$="R1R1R1R1R1"
1510 D$="R1R1R1R1R1"
1520 "!"
1530 A$="V13L4RG+8F+8G+AB1&B>D8C8DF2E8D8EA&"
1540 B$="V13>E1&EE8D8EF+G+2.AB2RA"
1550 C$="L8<G+>D<EDE>D<F+>D<G+>D<G+F+G+>D<A>D<B>D<BAB>D<A>D<G+>D
<G+F+G+>D<F>D"
1560 D$="R1R1R1R1"
1570 "!"
1580 A$="AG+8F+8G+ABB8A8B>C<D1&D2.R"
1590 B$="B>E2D8C8<BR1G+8F+8G+BE2.R"
1600 C$="<E>D<EDE>D<F+>D<G+>D<G+F+G+>D<A>D<B>D<BAB>D<A>D<G+>D<EF
+G+AB>C"
1610 "!"
1620 A$="V15L8<REF+G+AB>CDL4EEEDC2<B2"
1630 B$="V15R1>RC<BBA2G+2"
1640 C$="R1R4A4G+4E4E2E2"
1650 D$="DC<BAG+F+EDC<AB>CDEF+G+AB>CDE<BG+E"
1660 "!"
1670 A$=">D2DC<B2B>CDAG+AE2.E"
1680 B$="L8RAG+ARA4ARAG+ARA4AR4>C4<B4A4G+2.A4"
1690 C$="R1R1<F2.E4D2.C4"
1700 D$="F>FEF<F>F<E>E<DFEFD>D<E>E<RAG+A>D<A>C<ABAG+ABA>C<A"
1710 "!"
1720 A$="DC<B2R8BBB8A8G+8A>EC<B8A8AR2."
1730 B$="RA4ARAG+F+E4D4C2.>C4<A4G+4E4R2."
1740 C$=">D4R2.R1R4E4E4D4C4R2."
1750 D$="F4E4D4<B4G+4E4AB>CDEDCDE4<E4<A>AB>CDEFG"
1760 "!"
1770 A$="L8R2RFGAR>CDEFGAB->C4C4C4<B-4A2C2&"
1780 B$="<RB>RDE2B-2R2FR>A4G4G4F2E2"
1790 C$="R4C2.<G1AR>>F4E4C4C2G2"
1800 D$="A1>C2<B-2F<FGAB->CDEFGAB->CDEC"
1810 "!"
1820 A$="CV12<AFA>C<A>E-<A>D<B-FB->D<B->F<B->C<AFA>C<A>E-<A>D<B-
FB->D2&"
1830 B$="L4<FC<AF>FD<B-F>FC<AFB-2&B-8>C8<B-8A8"
1840 C$="R1R1R1R1"
1850 D$="R1R1R1R1"
1860 "!"
1870 A$="D<BGB>D<B>F<B>EC<G>CECGCD<BGB>D<B>F<B>EC<G>CECFC"
1880 B$="GD<BG>GEC<G>GD<BG>L8C>C<BAGFED"
1890 "!"
1900 A$="GECEGEB-EAFCFAF>C<FGECEGEB-EAFC<AF2&"
1910 B$="L4C<GEC>C<AFC>C<GECL8F2&FFGA"
1920 "!"
1930 A$="FAB->C<B-AGF>E-CD4.<B-AB->E<B->F<B->G<B-AGA>C<B-AB-AGA"
1940 B$="B->CDEFGAB->C<AFAB->DCDL4<GB->ECFC<A>C"
1950 "!"
1960 A$="L4FV15A>C<AFB-8>C8D<FGB->ECL8F<AGF>DC<B->C"
1970 B$="L8<FV12<AB->C<B-AGF>E-CD4.<B-AB->E<B->F<B->G<B-AGA>C<B-
AB-AGA"
1980 "!"
1990 A$="<A1&A>C<B-A>C<B-AB-G1&GB-AGB-AGAF1&FAGFAGFG"
2000 B$="V15>>RFE-DE-DCD<B-4R2.>RDC<B->DC<B->C<A4R2.R>C<B-A>C<B-
AB-G4R2."
2010 C$="<<<L4V12FV15A>CE-DFD<B-EGB->DCEC<ADFA>C<B->D<B-G"
2020 D$="R1R1R1R1R1R1"
2030 "!"
2040 A$="E2.E4F2R4A4B-2.G4A2.B4>C1R1"
2050 B$="R4G4>C4<B-4A4>C4F2.EDE2.DCD4F4.AGFGFEDCDEFGAB-G"
2060 C$="L8C16R16>C<B-AB-AGAC16R16B-AGAGFGC16R16AGFGFEFC16R16GFE
FEDEC16R16FEDEDCDEFGAB->CDE"
2070 "!"
2080 I=1:GOTO 170
2090 A$=LEFT$(A$,11)+"T190F>C<T160AT120G8F8F1&F1"
2100 B$=LEFT$(B$,24)+"1&C1"
2110 C$=LEFT$(C$,18)+"1&A1"
2120 D$="R1R1<<B-AGF>C4<C4F1&F1"
2130 "!"
2140 PLAY "T210P1";    :FOR I=0 TO K-1:PLAY A$(I);:NEXT
2150 PLAY ":P2K10R@4";:FOR I=0 TO K-1:PLAY A$(I);:NEXT
2160 PLAY ":P1";       :FOR I=0 TO K-1:PLAY B$(I);:NEXT
2170 PLAY ":P2K10R@4";:FOR I=0 TO K-1:PLAY B$(I);:NEXT
2180 PLAY ":P1";       :FOR I=0 TO K-1:PLAY C$(I);:NEXT
2190 PLAY ":P2K10R@4";:FOR I=0 TO K-1:PLAY C$(I);:NEXT
2200 PLAY ":P1";       :FOR I=0 TO K-1:PLAY D$(I);:NEXT
2210 PLAY ":P2K10R@4";:FOR I=0 TO K-1:PLAY D$(I);:NEXT
2220 PLAY ""
2230 IF F=1 THEN KEY0,"PLAY"+CHR$(&H22)+"X"
2240 END
2250 LABEL "!"
2260 A$(K)=A$:B$(K)=B$:C$(K)=C$:D$(K)=D$
2270 K=K+1
2280 RETURN
```

リスト4　代々木ゼミナール校歌


```
10 TEMPO 0:DEFSTR a-z:DEFINT L:' CHOICED VOICE.VTD ﾖｳ
20 '}g}g}g}g}g}g}g}g}g}g
30 PLAY "t96":' ｿﾎﾞｸﾆ ｱｶﾙｸ
40 '
50 b="c.c16defagf a4g4f2"
60 c="a.a16b->ccfed f4e4c2<"
70 d="f4fb-f>cc<a >c4.<b-a2"
80 e="f4b-gaf>cd c4<c4f4>f4<"
90 PLAY  "i18o4q8@v118L8p3k9 r1r1";
100 PLAY ":i18o4q8@v110L8p3k3 r1r1";
110 PLAY ":i1 o4q8@v118L8p1k2"+b;
120 PLAY ":i1 o4q8@v116L8p2k8"+b;
130 PLAY ":i1 o3q8@v116L8p1k7"+c;
140 PLAY ":i1 o3q8@v118L8p2k1"+c;
150 PLAY ":i1 o3q8@v120L8p3k6"+d;
160 PLAY ":i1 o2q8@v120L8p3k4"+e
170 '
180 a="c.c16defagc b-4afg4.r a.a16f>dc4<af c.a16ggf4.r"
190 b=a
200 c="a.a16b->ccfe<b- >g4fce2 f4fb-f4fc< a.>f16eec2<"
```

▶次号予告を見て、「うーん」とうなったのは，私だけではないはずだ。“代ゼミ校歌”ですか。うーん……。

吉田　哲 (19) 東京都

```
210 d="f4fb-f>cc<g >c4c<a>c2 c4cfc4c<a f.>c16c<b-a2"
220 e="f4b-gaf>cg e4f4c4>c4< f4ab-a4f4 c4c4<f4>f4<"
230 PLAY     a+":r16"+a+"16";
240 "bcde"
250 '
260 a="g.g16gab-ag4 a.a16>dcc4.r c.d16c<afag4 b-.b-16afg4.r"
270 b=a
280 c=">e.e16efgfe4 f4b-b-a2 f.f16fff4e4 g.g16f4e2<"
290 d="r1r1r1r1"
300 e=STRING$(6,"c>c<")+"f4>f4 a.b-16afaf>c4< e4fa>c4<c4<"
310 PLAY     a+":r16"+a+"16";
320 "bcde"
330 '
340 a="c.c16f4a.a16>c4< b-.a16gfg>c4c":'<
350 b=a+"<"
360 c="a.a16>c4f.f16f4 g.f16edee4e<"
370 d="f4f4>c4c4 c.c16c<a>c4<b-4"
380 e="f4a4>f4a4 e.f16cdc4c4<"
390 PLAY     a+"<"+":r16 "+a+"16<";
400 "bcde"
410 '
420 IF L=2 THEN 500
430 '
440 a="c.c16f2rr":b="c.c16f2.":c="a.a16>c2.<":d="f4a2.":e="f4f4>
c4<f4"
450 PLAY     a+":r16"+a+"16";
460 "bcde"
470 '
480 L=L+1:GOTO 180
490 '
500 a="c.c16f2r4 r1 r1"
510 b="c.c16f2>{cde}4 f4c4<b-.a16g>d c.<a16g4f2"
520 c="a.a16>c2r4 >c4<f4f4f4 f.f16e4c2<"
530 d="f4>c2r4< f4f4g4g4 a.>c16c<b-a2"
540 e="f4f4<f4>r4 f4a4g4b-4 t92>c4<t88c4t84f2t96"
550 PLAY     a+":r16"+a;
560 "bcde"
570 '
580 END
590 '
600 LABEL "bcde":PLAY ":"+b+":"+b;:PLAY ":"+c+":"+c;:PLAY ":"+d+
":"+e:RETURN
```

リスト5　サンダークロス

```
   1 /*                                                  sa."firstx
  10 /*             吾輩は、なんてったって
  20 /*
  30 /* 「さんだーくろす」の「First　Attack」
  40 /*
  50 /*                  なのである。
  60 /*
  70 /*               どうだ、まいったか。
  80 /*
  90 /* Programed　By 立川正之　in　89' 6/18
 100 /*
1000 dim char v(4,10)={
1010 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1020     61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1030 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1040     31,   0,   0,  15,   0,  35,   0,   2,   1,   0,   0,
1050     24,   0,   0,  15,   0,   2,   0,   2,   2,   0,   0,
1060     24,   0,   0,  15,   0,   2,   0,   2,   3,   0,   0,
1070     24,   0,   0,  15,   0,   2,   0,   4,   4,   0,   0}
1080 m_vset(1,v)
1090 /*
1100 v={
1110 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1120     61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1130 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1140     25,  10,   0,   7,   1,  32,   0,   2,   1,   0,   0,
1150     25,   0,   0,   8,   0,   2,   0,   2,   2,   0,   0,
1160     25,   0,   0,   8,   0,   2,   0,   2,   3,   0,   0,
1170     25,   0,   0,   8,   0,   2,   0,   4,   4,   0,   0}
1180 m_vset(2,v)
1190 /*
1200 v={
1210 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1220     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1230 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1240     31,   6,   2,   3,   5,  38,   0,  10,   3,   1,   0,
1250     31,  12,   8,   7,   5,   0,   0,   2,   3,   0,   0,
1260     31,   6,   2,   3,   5,  43,   0,   4,   7,   0,   0,
1270     31,  12,   8,   7,   5,   0,   0,   2,   7,   0,   0}
1280 m_vset(3,v)
1290 /*
1300 v={
1310 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1320     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1330 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1340     31,  13,   6,   8,   2,  30,   0,   0,   0,   0,   0,
1350     20,  14,   6,   8,   3,  20,   2,   5,   0,   0,   0,
1360     31,   5,   5,   8,   4,  40,   0,   0,   0,   0,   0,
1370     31,  13,   6,   8,   5,   0,   0,   1,   0,   0,   0}
1380 m_vset(4,v)
1390 /*
1400 v={
1410 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1420     53,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1430 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1440     31,  24,  25,  15,  10,   0,   0,   2,   0,   1,   0,
1450     31,  19,  18,  15,  10,   0,   0,   1,   0,   0,   0,
1460     31,  19,  18,  15,  10,   0,   0,   1,   0,   0,   0,
1470     31,  19,  18,  15,  10,   0,   0,   1,   0,   0,   0}
1480 m_vset(5,v)
1490 /*
1500 v={
1510 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1520     56,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1530 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1540     31,   1,   5,   4,   2,   0,   0,  14,   0,   3,   0,
1550     23,   9,   6,   4,   0,   0,   1,  12,   0,   2,   0,
1560     31,   1,   6,   4,   0,   0,   0,   2,   0,   1,   0,
1570     31,  21,  19,  15,   3,   0,   1,   2,   0,   0,   0}
1580 m_vset(6,v)
1590 /*
1600 v={
1610 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1620     37,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1630 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1640     31,  26,  25,  15,   7,   0,   0,   8,   0,   0,   0,
1650     31,  19,  18,  15,   7,   0,   0,   1,   0,   0,   0,
1660     31,  19,  17,  15,   5,   0,   0,   1,   0,   0,   0,
1670     31,  15,  15,  15,   5,   0,   0,   0,   0,   0,   0}
1680 m_vset(7,v)
1690 /*
1700 v={
1710 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1720     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1730 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1740     15,  11,   0,   6,   1,  26,   1,   2,   3,   0,   0,
1750     15,  11,   0,   6,   1,  33,   1,   2,   0,   0,   0,
1760     15,  11,   0,   6,   1,  38,   1,   2,   7,   0,   0,
1770     22,   0,   0,   7,   0,   0,   0,   2,   0,   0,   0}
1780 m_vset(8,v)
1790 /*
1800 v={
1810 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1820     61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1830 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1840     16,   8,   0,   8,   1,  27,   0,   2,   0,   0,   0,
1850     24,   3,   0,   8,   1,   0,   0,   2,   0,   0,   0,
1860     24,   3,   0,   8,   1,   0,   0,   2,   0,   0,   0,
1870     24,   3,   0,   8,   1,   0,   0,   2,   0,   0,   0}
1880 m_vset(9,v)
1890 /*
1900 v={
1910 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
1920     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
1930 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
1940     31,   0,   0,   0,   0,  30,   0,   8,   3,   0,   0,
1950     20,   0,   0,   8,   0,   4,   0,   8,   3,   0,   0,
1960     31,   0,   0,   0,   0,  30,   0,   8,   7,   0,   0,
1970     20,   0,   0,   8,   0,   4,   0,   8,   7,   0,   0}
1980 m_vset(10,v)
1990 /*
2000 v={
2010 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
2020     29,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
2030 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
2040     31,  26,  25,  15,   5,   0,   0,  10,   0,   0,   0,
2050     31,  17,  16,  15,   5,   8,   0,   2,   0,   0,   0,
2060     31,  17,  16,  15,   5,   8,   0,   2,   0,   0,   0,
2070     31,  17,  16,  15,   5,   8,   0,   2,   0,   0,   0}
2080 m_vset(11,v)
2090 /*
9000 m_init():for i=1 to 8:m_alloc(i,3000):next
9010 for i=1 to 8:m_assign(i,i):next
9020 str a[256],b[256],c[256],d[256],e[256],f[256]
9030 str g[256],h[256],j[256],k[256],l[256],m[256]
9040 str n[256],o[256],p[256],q[256],r[256],s[256]
9050 str t[256],u[256],w[256],x[256],y[256],z[256]
9060 str aa[256],bb[256],cc[256],dd[256],ee[256],ff[256]
9070 str gg[256],hh[256],jj[256],kk[256],ll[256],mm[256]
9080 str nn[256],oo[256],pp[256],qq[256],rr[256],ss[256]
9090 str tt[256],uu[256],ww[256],xx[256],yy[256],zz[256]
9100 str sd[256],bd[256],hc[256],t1[256],t2[256],t3[256]
9110 sd="y15,253@7y2,14<c>y15,0
9120 bd="@5y2,2c
9130 hc="@6y2,65<<c>>
9140 t1="y2,29g
9150 t2="y2,30e
9160 t3="y2,31c
10000 /*
10010 /*
10020 a="t120  [d.c.] [coda] o3 l16  @4 p3 q8 @v127 y48,40
10030 b="|:8dddd dddd dddd dddd:|
10040 c="|:4dddd dddd dddd dddc:|
10050 d="|:4ffff ffff ffff ffff:|>
10060 e="|:2gggg gggg gggg gggg:|
10070 f="|:6aaaa:|aab-b-bb<cc>
10080 g="|:4b-b-b-b-:|<|:4cccc:|>|:4b-b-b-b-:|<|:11f:|c8>b-ab-
10090 h="|:4b-b-b-b-:|<|:11c:|>g<ceg<c>|:6dddd:|<dc>bagf+ed
10100 z="[*]
11020 m_trk(1,a) /*  あ～、なんか気分がいいのである。
11030 m_trk(1,b) /*なぜかってーと、ここが妙に居心地が良いからで
11040 m_trk(1,c) /*ある。
11050 m_trk(1,d) /*  いや～実にいい気持ちだ。
11060 m_trk(1,e) /*  こんなスペースを見付けてしまった自分が凄い
11070 m_trk(1,f) /*と思うな。うへへ。
11080 m_trk(1,g) /*  まぁ、そんなことはどうでもいい。ここは私の
11090 m_trk(1,h) /*場所だ。誰にも文句はいわせないぞ。
11100 m_trk(1,z) /*  えっへん。
12000 /*
12010 /*
12020 a="       [d.c.] [coda] o3 @12  @1 p3 q8   v14 y49,20
12030 bb="y49,020c& y49,050c& y49,080c& y49,110c& y49,140c& y49,
170c& y49,200c& y49,230c&
12040 cc="y49,020c+&y49,050c+&y49,080c+&y49,110c+&y49,140c+&y49,
170c+&y49,200c+&y49,230c+&
12050 dd="y49,020d& y49,050d& y49,080d& y49,110d& y49,140d& y49,
170d& y49,200d& y49,230d&
12060 ee="y49,020d+&y49,050d+&y49,080d+&y49,110d+&y49,140d+&y49,
170d+&y49,200d+&y49,230d+&
12070 ff="y49,020e& y49,050e& y49,080e& y49,110e& y49,140e& y49,
170e& y49,200e& y49,230e&
12080 gg="y49,020f& y49,050f& y49,080f& y49,110f& y49,140f& y49,
```

▶僕は麻雀悟空で副落率0.490で七段だ。哭きの竜と呼んでくれ。多分このへんが副落率5割近くを保ちながら昇段できる限界だと思うのだが……。私は誰の挑戦でも受ける。

堀田　裕志（18）滋賀県

```
170f& y49,200f& y49,230f&
12090 hh="y49,020f+&y49,050f+&y49,080f+&y49,110f+&y49,140f+&y49,
170f+&y49,200f+&y49,230f+&
12100 jj="y49,020g& y49,050g& y49,080g& y49,110g& y49,140g& y49,
170g& y49,200g& y49,230g&
12110 kk="y49,020g+&y49,050g+&y49,080g+&y49,110g+&y49,140g+&y49,
170g+&y49,200g+&y49,230g+&
12120 ll="y49,020a& y49,050a& y49,080a& y49,110a& y49,140a& y49,
170a& y49,200a& y49,230a&
12130 mm="y49,020a+&y49,050a+&y49,080a+&y49,110a+&y49,140a+&y49,
170a+&y49,200a+&y49,230a+&
12140 nn="y49,020b& y49,050b& y49,080b& y49,110b& y49,140b& y49,
170b& y49,200b& y49,230b&<
12150 b="y49,20d2.&|1c1@2q8>b8.g8.a2&a8:|
12160 c="@9116>>|:2q7d8.q8a4.rq5gfq6g8q5fefedq6e8q5dq6c8|1q8d4..
r:|q7d8.q5cdefg
12170 d="|:2q7f8.<q8c4.rq5>b-aq6b-8q5agagfq6g8q5fq6e8|1q8f4..r:|
q7f8.q5>a<cfa<c>q7g8.b8.<d8.&@12
12180 e="|:3y49,180d-&y49,090d-&y49,180d-&y49,20d&y49,110d&y49,2
00d&y49,110d&y49,20d&:|y8,1
12190 f="116q5c>bgq7g8.b8.q8<d4>>q5gb<dgb<d>q7a8.<c+8.e8.&@12
12200 g="|:3y49,180e-&y49,090e-&y49,180e-&y49,20e&y49,110e&y49,2
00e&y49,110e&y49,20e&:|y8,1
12210 h="116q5c+eaq7e8.a8.<c+8&@12
12220 j="|:6y49,180c&y49,090c&y49,180c&y49,20c+&y49,110c+&y49,20
0c+&y49,110c+&y49,20c+&:|y8,1
12230 k="116>q7   f8.e8.d4d8q5defgq7e8.c8.>g8&@12
12240 l="|:6y49,180g-&y49,090g-&y49,180g-&y49,20g&y49,110g&y49,2
00g&y49,110g&y49,20g&:|y8,1
12250 m="116q7<f8.e8.d4d8q5defgq7a8.f8.c8q5afc<q6c8q5>b-ab-
12260 n="q7f8.e8.d4d8q5defg<q7c8.>g8.e4&e16>q5g<ceg<cq8d4&@12
12270 o="|:21y49,180d-&y49,090d-&y49,180d-&y49,20d&y49,110d&y49,
200d&y49,110d&y49,20d&:|y8,1
13020 m_trk(2,a)/*そんでもって、問題はここに何を書くかである。
13030 m_trk(2,"y49,20|:2d4&"+dd)/*（←お前じゃまだよ）
13040 m_trk(2,ee)/*　要するにこのスペースはプログラムを打ち込む
13050 m_trk(2,ff)/*人しか読まないわけだから私の戯言をばんばん書
13060 m_trk(2,gg)/*いても平気だなとか思ってたのに、編集の方に聞
13070 m_trk(2,hh)/*いてみるとやっぱり「Ｈ」な話はだめだそうだ。
13080 m_trk(2,jj)/*　あ～あ。
13090 m_trk(2,kk)/*　内緒にしとけば良かったな。
13100 m_trk(2,ll)/*　仕方がない。
13110 m_trk(2,mm)/*　そんなわけでここで「しけべ」な話ができなく
13120 m_trk(2,nn)/*なってしまったので、突然「立川くんのよいこの
13130 m_trk(2,bb)/*ためのＦＭ音源講座」みたいなもの（と言うあた
13140 m_trk(2,cc)/*りがみそ）をやってしまおうということにした。
13150 m_trk(2,b)/*　しかし、ちゃんとした企画ではないので内容が
13160 m_trk(2,"y49,20<c4.&@l1"+bb)/*（←あ～、また邪魔をする）
13170 m_trk(2,cc)/*おちゃらけてしまうのは勘弁していただきたい。
13180 m_trk(2,dd)/*でも、他では聞けない美味しい話を少しずつ、出
13190 m_trk(2,ee)/*しおしみをしながら書いてくつもりなので、結構
13200 m_trk(2,ff)/*役には立つはずだ。ここはひとつ、友達には内緒
13210 m_trk(2,gg)/*で、トイレなんかで「うふふ」と薄笑いしながら
13220 m_trk(2,hh)/*読むのがいいだろう。一応、建て前としては「紙
13230 m_trk(2,jj)/*面の余分なスペースを利用した合理的で素晴らし
13240 m_trk(2,kk)/*い講座」ということになっている。かっかっか。
13250 m_trk(2,ll)/*　あんまり馬鹿なことばっかり書いてると本当に
13260 m_trk(2,mm)/*馬鹿だと思われるな。
13270 m_trk(2,nn)/*　そろそろ真面目なふりをしよう。
13280 m_trk(2,bb)/*　では、「立川くんのよいこのためのＦＭ音源講
13290 m_trk(2,cc)/*座～第一回～」をはじめることにする。
13300 m_trk(2,dd)/*　そうそう、題から見ても分かるとおり、「わる
13310 m_trk(2,ee)/*いこ」は読んではいけないんだぞ。ざまーみろ。
13320 m_trk(2,ff)/*　んでもって第一回の今回はなにをやろうかなと
13330 m_trk(2,gg)/*うことになるのだが、いきなり最近流行の「Ｙコ
13340 m_trk(2,hh)/*マンドずらずら」の技を解説することにしよう。
13350 m_trk(2,jj)/*　まず、12030行から12140行辺りを見て頂こう。
13360 m_trk(2,kk+"y49,20a2.")/*（←お願いだから邪魔しないで）
13370 m_trk(2,c) /*　うんざりするほど「Ｙコマンド」が並んでいる
13380 m_trk(2,d) /*はずだ。これがなにを意味しているかというと、
13390 m_trk(2,e) /*聞いてもわかるとおり、「ポルタメント」を再現
13400 m_trk(2,f) /*しているのである。「Ｙコマンド」はＦＭ音源の
13410 m_trk(2,g) /*レジスタに直接数値を書き込む命令だが、ここで
13420 m_trk(2,h) /*は「すこしだけ周波数をずらす」ための“技”に
13430 m_trk(2,j) /*使っている。なぜこんなことが必要になるかとい
13440 m_trk(2,k) /*うと、１オクターブのポルタメントをかなり長い
13450 m_trk(2,l) /*時間をかけて行うので、普通に音階を「＆」でつ
13460 m_trk(2,m) /*なげただけでは音が粗くなってしまい、聞いた耳
13470 m_trk(2,n) /*が（反語：見た目）あまり良くないからである。
13480 m_trk(2,o) /*そんなことになるくらいならばプログラムしない
13490 m_trk(2,z) /*方がまだましだというポリシーを私は持っている
14000 /*           ＧＯＴＯ　１５０２０
14010 /*
14020 a=" r32. [d.c.] [coda] o3 @12  @1 p3 q8   v11 y50,00
14030 bb="y50,000c& y50,030c& y50,060c& y50,090c& y50,120c& y50,
150c& y50,180c& y50,210c&
14040 cc="y50,000c+&y50,030c+&y50,060c+&y50,090c+&y50,120c+&y50,
150c+&y50,180c+&y50,210c+&
14050 dd="y50,000d& y50,030d& y50,060d& y50,090d& y50,120d& y50,
150d& y50,180d& y50,210d&
14060 ee="y50,000d+&y50,030d+&y50,060d+&y50,090d+&y50,120d+&y50,
150d+&y50,180d+&y50,210d+&
14070 ff="y50,000e& y50,030e& y50,060e& y50,090e& y50,120e& y50,
150e& y50,180e& y50,210e&
14080 gg="y50,000f& y50,030f& y50,060f& y50,090f& y50,120f& y50,
150f& y50,180f& y50,210f&
14090 hh="y50,000f+&y50,030f+&y50,060f+&y50,090f+&y50,120f+&y50,
150f+&y50,180f+&y50,210f+&
14100 jj="y50,000g& y50,030g& y50,060g& y50,090g& y50,120g& y50,
150g& y50,180g& y50,210g&
14110 kk="y50,000g+&y50,030g+&y50,060g+&y50,090g+&y50,120g+&y50,
150g+&y50,180g+&y50,210g+&
14120 ll="y50,000a& y50,030a& y50,060a& y50,090a& y50,120a& y50,
150a& y50,180a& y50,210a&
14130 mm="y50,000a+&y50,030a+&y50,060a+&y50,090a+&y50,120a+&y50,
150a+&y50,180a+&y50,210a+&
14140 nn="y50,000b& y50,030b& y50,060b& y50,090b& y50,120b& y50,
150b& y50,180b& y50,210b&<
14150 b="y50,00d2.&|1c1@2q8>b8.g8.a2&a8:|
14180 e="|:3y50,160d-&y50,070d-&y50,160d-&y50,00d&y50,090d&y50,1
80d&y50,090d&y50,00d&:|y8,2
14200 g="|:3y50,160e-&y50,070e-&y50,160e-&y50,00e&y50,090e&y50,1
80e&y50,090e&y50,00e&:|y8,2
14220 j="|:6y50,160c&y50,070c&y50,160c&y50,00c+&y50,090c+&y50,18
0c+&y50,090c+&y50,00c+&:|y8,2
14230 l="|:6y50,160g-&y50,070g-&y50,160g-&y50,00g&y50,090g&y50,1
80g&y50,090g&y50,00g&:|y8,2
14240 k="116>q7v12f8.e8.d4d8q5defgq7e8.c8.>g8&@12
14250 o="|:21y50,160d-&y50,070d-&y50,160d-&y50,00d&y50,090d&y50,
180d&y50,090d&y50,00d&:|y8,2
15020 m_trk(3,a) /*ので、だれかのいうところの“清く正しくズリズ
15030 m_trk(3,"y50,00|:2d4&"+dd)/* リと”「Ｙコマンド」を並べた
15040 m_trk(3,ee)/*のである。なに？　「音程と長さを指定すればＭ
15050 m_trk(3,ff)/*ＭＬが返ってくるようなサブルーチンを作ればい
15060 m_trk(3,gg)/*いじゃん」だって？　まさにそのとおり。しかし
15070 m_trk(3,hh)/*私のもうひとつのポリシーに「見ただけで鼻血が
15080 m_trk(3,jj)/*つう、と滑り降りてくるようなうんざりするプロ
15090 m_trk(3,kk)/*グラム」を組む、というのがあって、仕方の無い
15100 m_trk(3,ll)/*ことなのである。別に「胡散臭いリストを打ち終
15110 m_trk(3,mm)/*えたあとの充実感」とかそんなものを狙っている
15120 m_trk(3,nn)/*わけでは無いので安心して頂きたい。
15130 m_trk(3,bb)/*　んっと、話がそれた。
15140 m_trk(3,cc)/*　「Ｙコマンド」の話だったね。
15150 m_trk(3,b) /*　では、実際に「周波数ずらし」をするのにどう
15160 m_trk(3,"y50,00<c4.&@l1"+bb)/*すればいいか？　説明しよう。
15170 m_trk(3,cc)/*　まず、「Ｙなんとか，かんとか」という書式で
15180 m_trk(3,dd)/*指定するのだが、“なんとか”の部分にはチャン
15190 m_trk(3,ee)/*ネル番号を、“かんとか”の部分には実際にどの
15200 m_trk(3,ff)/*程度周波数をずらすか書き込む。でもってチャン
15210 m_trk(3,gg)/*ネル番号は「１～８」ではなく、「４８～５５」
15220 m_trk(3,hh)/*の値で指定するので注意が必要である。周波数の
15230 m_trk(3,jj)/*ずらし具合いは「０～２５５」までの値で指定し
15240 m_trk(3,kk)/*て、０で無変化、２５５で上に半音ずれる（ほん
15250 m_trk(3,ll)/*とは少し違うんだけどそう考えても構わない）。
15260 m_trk(3,mm)/*つまり、「Ｙ５１，１２８」というのは「第４チ
15270 m_trk(3,nn)/*ャンネルを上に半音の半音程ずらすんだよ」とい
15280 m_trk(3,bb)/*うことなのだ。
15290 m_trk(3,cc)/*　この「極意、周波数ずらしの法」はなかなかど
15300 m_trk(3,dd)/*うして使える奴で、ポルタメントの他、２チャン
15310 m_trk(3,ee)/*ネル重ねて音を出す時に音色を太くするためとか
15320 m_trk(3,ff)/*「ソフトＬＦＯ」を得るときなど、様々なとこで
15330 m_trk(3,gg)/*活躍している。詳しくは本文にて述べているので
15340 m_trk(3,hh)/*そちらを読んで頂くことにしよう。
15350 m_trk(3,jj)/*　さて、この「Ｙコマンド」にはもう一つの活用
15360 m_trk(3,kk+"y50,00a2.")/*法がある。ノイズジェネレータを使
15370 m_trk(3,c) /*かうためのものである。このノイズジェネレータ
15380 m_trk(3,d) /*は文字どおりノイズを発生するためのもので、今
15390 m_trk(3,e) /*回のプログラムではスネアドラムに使用した。
15400 m_trk(3,f) /*　ＣＤで聞いてみたところ、この「サンタクロー
15410 m_trk(3,g) /*ス」もとい「サンダークロス」はスネアドラムの
15420 m_trk(3,h) /*アタック音が強く、なかなか存在感のある音色だ
15430 m_trk(3,j) /*ったのでありがちな「アルゴリズム４番の第１、
15440 m_trk(3,k) /*２オペレータでノイズを発生し、第３、４オペレ
15450 m_trk(3,l) /*ータでトーンを発生する」という手法はあえてと
15460 m_trk(3,m) /*らずにアルゴリズム５番で、第１、２、３オペレ
15470 m_trk(3,n) /*ータでトーン、第４オペレータ（ノイズジェネレ
15480 m_trk(3,o) /*ータ）でノイズをだすという、大道芸をやってし
15490 m_trk(3,z) /*まっている。
16000 /*           ＧＯＴＯ　１７０２０
16010 /*
16020 a=" r16. [d.c.] [coda] o3 @12  @1 p3 q8    v9 y51,40
16030 bb="y51,040c& y51,070c& y51,100c& y51,130c& y51,160c& y51,
190c& y51,220c& y51,250c&
16040 cc="y51,040c+&y51,070c+&y51,100c+&y51,130c+&y51,160c+&y51,
190c+&y51,220c+&y51,250c+&
16050 dd="y51,040d& y51,070d& y51,100d& y51,130d& y51,160d& y51,
190d& y51,220d& y51,250d&
16060 ee="y51,040d+&y51,070d+&y51,100d+&y51,130d+&y51,160d+&y51,
190d+&y51,220d+&y51,250d+&
16070 ff="y51,040e& y51,070e& y51,100e& y51,130e& y51,160e& y51,
190e& y51,220e& y51,250e&
16080 gg="y51,040f& y51,070f& y51,100f& y51,130f& y51,160f& y51,
190f& y51,220f& y51,250f&
16090 hh="y51,040f+&y51,070f+&y51,100f+&y51,130f+&y51,160f+&y51,
190f+&y51,220f+&y51,250f+&
16100 jj="y51,040g& y51,070g& y51,100g& y51,130g& y51,160g& y51,
190g& y51,220g& y51,250g&
16110 kk="y51,040g+&y51,070g+&y51,100g+&y51,130g+&y51,160g+&y51,
190g+&y51,220g+&y51,250g+&
16120 ll="y51,040a& y51,070a& y51,100a& y51,130a& y51,160a& y51,
190a& y51,220a& y51,250a&
16130 mm="y51,040a+&y51,070a+&y51,100a+&y51,130a+&y51,160a+&y51,
190a+&y51,220a+&y51,250a+&
16140 nn="y51,040b& y51,070b& y51,100b& y51,130b& y51,160b& y51,
190b& y51,220b& y51,250b&<
16150 b="y51,40d2.&|1c1@2q8>b8.g8.a2&a8:|
16180 e="|:3y51,200d-&y51,110d-&y51,200d-&y51,40d&y51,130d&y51,2
20d&y51,130d&y51,40d&:|y8,3
16200 g="|:3y51,200e-&y51,110e-&y51,200e-&y51,40e&y51,130e&y51,2
20e&y51,130e&y51,40e&:|y8,3
16220 j="|:6y51,200c&y51,110c&y51,200c&y51,40c+&y51,130c+&y51,22
0c+&y51,130c+&y51,40c+&:|y8,3
16230 l="|:6y51,200g-&y51,110g-&y51,200g-&y51,40g&y51,130g&y51,2
20g&y51,130g&y51,40g&:|y8,3
16240 k="116>q7v10f8.e8.d4d8q5defgq7e8.c8.>g8&@12
16280 o="|:21y51,200d-&y51,110d-&y51,200d-&y51,40d&y51,130d&y51,
220d&y51,130d&y51,40d&:|y8,3
17020 m_trk(4,a) /*　こうすることによって、トーンの部分で表現す
17030 m_trk(4,"y51,40|:2d4&"+dd) /*るアタック感に他のパートの音
17040 m_trk(4,ee)/*に負けない迫力をもたせることができるわけなの
17050 m_trk(4,ff)/*である。もちろん、これはＯＰＭＡを使ってサン
17060 m_trk(4,gg)/*プリングドラムで聞く人にはあまり意味のないこ
17070 m_trk(4,hh)/*とだが、そうでない人にとっては重大な問題なの
17080 m_trk(4,jj)/*だから馬鹿にしてはいけないぞ。
17090 m_trk(4,kk)/*　ではこれを「Ｙコマンド」行うにはどうするか
17100 m_trk(4,ll)/*説明しよう。今度は「Ｙ１５，なんとか」という
17110 m_trk(4,mm)/*形で設定をする。こうすると自動的に第８チャン
17120 m_trk(4,nn)/*ネルの第４オペレータがノイズを出力するように
17130 m_trk(4,bb)/*なるのだ。また、エンベロープはこの３２スロッ
17140 m_trk(4,cc)/*ト目のエンベロープ機能をそのまま利用して設定
17150 m_trk(4,b) /*するものの、このときのエンベロープ変化は対数
17160 m_trk(4,"y51,40<c4.&@l1"+bb) /*的変化ではなくアタック部は
17170 m_trk(4,cc)/*エクスポーネンシャル、ディケイ部は全て直線的
```

▶やっと福袋が手に入ったぞぉー！とよろこんで家に帰ってパッケージを開けると，「おおっ，マニュアルが厚いっ！」しかしながら，Cはもっとマニュアルが厚いことを知って驚きにけり。ちまちまとプログラムをつくるとしよう。へへへ。　宮原　大 (16) 長野県

```
17180 m_trk(4,dd)/*に変化するので気を付ける必要がある。とかなん
17190 m_trk(4,ee)/*とかいって知ったかぶりをしていても、実は私も
17200 m_trk(4,ff)/*よーわからんのであんまり気にすることはないと
17210 m_trk(4,gg)/*思うわけだな。けけけけけ。
17220 m_trk(4,hh)/*　そして、実際に“なんとか”に代入する値だが
17230 m_trk(4,jj)/*８ビットデータで上位１ビットがノイズのＯＮ、
17240 m_trk(4,kk)/*ＯＦＦを、下位５ビットで「ノイズ周波数」を操
17250 m_trk(4,ll)/*る。この場合だと、ノイズをＯＮにして、周波数
17260 m_trk(4,mm)/*を設定するということなので、十進数でとる値は
17270 m_trk(4,nn)/*「１２９～１５９」までとなる。しかしながらう
17280 m_trk(4,bb)/*さんくさい私は「２２５～２５５」を使う……
17290 m_trk(4,cc)/*　まぁ、そんなことはどうでもいい。
17300 m_trk(4,dd)/*　あ、そうそう、ノイズの周波数というのはその
17310 m_trk(4,ee)/*ノイズに含まれる周波数帯の高低を表している。
17320 m_trk(4,ff)/*つまり、周波数の低いノイズはいわゆる「ピンク
17330 m_trk(4,gg)/*ノイズ」で、「ざぁ～～」とか「ごぉ～～」とか
17340 m_trk(4,hh)/*いったりするし、周波数の高いのは「ホワイトノ
17350 m_trk(4,jj)/*イズ」で「さぁ～～」とかいうのである。時と場
17360 m_trk(4,kk+"y51,40a2.")/*合によって使い別けて貰いたい。
17370 m_trk(4,c) /*　さて、今回の「立川くんのよいこのためのＦＭ
17380 m_trk(4,d) /*音源講座」はそろそろ終わりにしよう。行数も残
17390 m_trk(4,e) /*り少なくなってきたしね。
17400 m_trk(4,f) /*　一応、第一回ということで初心者向けの簡単な
17410 m_trk(4,g) /*内容でまとめてみたが（まとまってないって）い
17420 m_trk(4,h) /*かがなもんだろう。
17430 m_trk(4,j) /*　誰だ！！　内容が無いよう、といってるのは。
17440 m_trk(4,k) /*　ま、軽い気持ちで書き出したのに、正直言って
17450 m_trk(4,l) /*もう一本プログラムを組む位の労力を使ってしま
17460 m_trk(4,m) /*ったので、第二回はやれるかどうかわからない。
17470 m_trk(4,n) /*　原稿料２倍もらえるわけではないしね。
17480 m_trk(4,o) /*　しかし、可能性がないわけではないような気が
17490 m_trk(4,z) /*する。ま、とりあえずは、お・ち・ま・い。
18000 /*
18010 /*
18020 a="      [d.c.] [coda] o3 l16  @3 p3 q8   v12 y52,00
18030 b="|:19cd<f>fg<g>g<a>g<g>fg<fed>a:|
18040 c="l32>b<cc+dd+eff+gg+aa+b<cc+dd+eff+gg+aa+b<cc+dd+eff+l16
18050 d="|:5fd>b-<:|f|:2gec:|ger>cef<cea<c>|:5fd>b-<:|f|:2afc:|a
fafc<c8>b-ab-
18060 e="|:5fd>b-<:|f|:3gec:|ge>g<ceg<cd>af+af+df+a<d>af+d>af+d>
a
18070 f="l32cc+dd+eff+gg+aa+b<cc+dd+eff+gg+aa+b<cc+dd+eff+g
19020 m_trk(5,a) /*　そうそう、余談になるが、６月号で載せた「フ
19030 m_trk(5,b) /*ーガ」が意外に好評で、アンケート葉書に良かっ
19040 m_trk(5,c) /*たというメッセージを書いてくれたひとが結構沢
19050 m_trk(5,d) /*山いた。なかには「質問電話」にわざわざそれを
19060 m_trk(5,e) /*言うために電話をかけてきたお父さんもいたとい
19070 m_trk(5,f) /*うのだ。
19080 m_trk(5,z) /*　たいへん恐縮です。
20000 /*
20010 /*
20020 a="      [d.c.] [coda] o3 l16  @8 p3 q8   v15 y53,10
20030 b="a1<c1d1c2>b2a1<c1d1d1
20040 c="@10>>d1&d1d1&d1f1&f1f1&f1
20050 d="|:2d8.d8.d8&d2:||:2e8.e8.e8&e2:|
20060 e="b-1<c1>b-1<f2r2>b-1<c2&c8.r4r16d1&d1
21020 m_trk(6,a) /*　自分ではエンディングが突然モノラルになって
21030 m_trk(6,b) /*しまうのが嫌だったのだが、それは単なる言い訳
21040 m_trk(6,c) /*にしかならん。とにかく、入力してくれた人、
21050 m_trk(6,d) /*　ありがとう。
21060 m_trk(6,e) /*　そんなこともあって、今回のこのプログラムは
21070 m_trk(6,z) /*俄然、気合いが入ってしまったのだ。
22000 /*
22010 /*
22020 a="      [d.c.] [coda] o4 l16  @8 p3 q8   v15 y54,30
22030 b="d1f1g1f2e2d1f1g1a1
22040 c="@10>>a1&a1a1&a1<c1&c1c1&c1>
22050 d="|:2g8.g8.g8&g2:||:2a8.a8.a8&a2:|
22060 e="f1g1f1<c2r2>f1g2&g8.r4r16a1&a1
23020 m_trk(7,a) /*なにせ、制作時間もこれまでの倍位かけたし（土
23030 m_trk(7,b) /*日をまるまるつかってしまった）、いろんな人に
23040 m_trk(7,c) /*聞いてもらってアドバイスももらった。お陰でど
23050 m_trk(7,d) /*こへ出しても恥ずかしくないデキになったのさ。
23060 m_trk(7,e) /*　だれからの挑戦も受けてやるぞ。
23070 m_trk(7,z) /*　挑戦するなら送ってくるんだ。
24000 /*
24010 /*
24020 a="      [d.c.] [coda] o2 l16  @8 p3 q8 @v127 y55,20 y3,3
24030 b="|:7"+bd+hc+sd+hc+bd+"c"+sd+hc+bd+hc+sd+bd+hc+bd+sd+hc+"
:|
24040 c="|:16"+sd+":|"
24050 d="|:8"+bd+hc+sd+hc+bd+"c"+sd+hc+bd+hc+sd+bd+hc+bd+sd+hc+"
:|
24060 e="|:3"+sd+bd+"c"+sd+bd+"c"+sd+bd+"cc"+sd+bd+"c32c32c32c32
"+sd+bd+":|
24070 f=bd+"|:7"+sd+":|@11l32|:4"+t1+":||:4"+t2+":||:4"+t3+":||:
4"+t2+":|l16
24080 n=bd+hc+sd+hc+bd+"c"+sd+hc+bd+hc+sd+bd+hc+bd+sd+hc
24090 g=n+bd+hc+sd+hc
24100 h=bd+"c"+sd+hc+"@11|:2"+t1+":||:2"+t2+":||:2"+t3+":||:2"+t
2+":|"
24110 j=n+bd+hc+sd+hc
24120 k=bd+"c"+sd+hc+"|:4"+sd+":|@11l32|:2"+t1+":||:2"+t2+":||:2
"+t3+":||:2"+t2+":|l16
24130 l=n+bd+hc+sd+hc
24140 m=bd+"c"+sd+hc+bd+hc+sd+bd+t3+t2+t1+t2
24150 n=n+bd+"|:7"+sd+":|@11l32
24160 o="|:4"+t1+":||:4"+t2+":||:4"+t3+":||:4"+t2+":|l16
25020 m_trk(8,a) /*　待ってるぞ。
25030 m_trk(8,b) /*　もしも、私の鼻をあかしたなら、土下座でもな
25040 m_trk(8,c) /*んでもしようじゃないか。
25050 m_trk(8,d) /*　とまあ、ついつい、ごたいそうなことを書いて
25060 m_trk(8,e) /*しまったよ。いや、ついついキーボードが滑って
25070 m_trk(8,f) /*しまったのさ。
25080 m_trk(8,g) /*　勘弁して下さい。
25090 m_trk(8,h) /*　ともかく一生懸命作ったのだから一生懸命打ち
25100 m_trk(8,j) /*込んでほしいと思うばかりなのだ。もしも、入力
25110 m_trk(8,k) /*したなら、なんらかの反響が欲しいので、アンケ
25120 m_trk(8,l) /*ートの隅にでも書いて下さいな。
25130 m_trk(8,m) /*　もちろん、厳しい意見、リクエストもお待ちし
25140 m_trk(8,n) /*てます。「講座を続けてほしいな」というひとが
25150 m_trk(8,o) /*ればそう書いてね。ひょっとすると、きまぐれを
25160 m_trk(8,z) /*起こして書くかもしんないから。
60000 /*          ではまた、さよなら、さよなら、ｻﾖﾅﾗ､ｻﾖﾅﾗ･･･
60010 /*
60020 m_play()
```

リスト6　ソーサリアン

```
10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
20 /*              S o r c e r i a n                 /*
30 /*                                                /*
40 /*      呪われたクイーンマリー号（船内のテーマ）      /*
50 /*                                                /*
60 /*                  (c) Falcom                    /*
70 /*                                                /*
80 /*          Arranged by Z.Nishikawa               /*
90 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
100   dim     char    v(4,10)={        /*     f l u t e
110 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
120       59, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
130 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
140       31,  8,  2,  2, 14, 40,  3,  6,  0,  0,0,
150       19,  6,  0,  2, 15, 51,  1,  6,  3,  0,0,
160       15, 10,  0,  2,  2, 38,  3,  3,  6,  0,0,
170       14,  3,  0,  3,  0,  0,  2,  3,  3,  0,0}
180   m_vset(71,v)
190   vset2(91)
200   v={               /*      b a c k i n g  p i a n o
210 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
220       60, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
230 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
240       31,  0,  0,  0,  0, 33,  0,  8,  7,  0,0,
250       17, 14,  8,  3,  3,  8,  0,  8,  7,  0,0,
260       31,  0,  0,  0,  0, 24,  0,  4,  3,  0,0,
270       31, 14,  8,  3,  3,  0,  0,  4,  3,  0,0}
280   m_vset(72,v)
290   v={               /*      b o t t l e  b l o w
300 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
310       62, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
320 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
330        3,  0,  2,  4,  1, 55,  3,  4,  6,  0,0,
340        3,  7,  2,  4,  1, 37,  3,  2,  6,  0,0,
350        3,  0,  2,  4,  1, 10,  3,  2,  6,  0,0,
360        3,  0,  2,  4,  1,  0,  0,  2,  6,  0,0}
370   m_vset(73,v)
380   vset2(93)
390   v={               /*      s y n t h e  l e a d
400 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
410        6, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
420 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
430       31, 10,  0,  2, 15,  8,  0,  4,  4,  0,0,
440       31, 10,  0,  1,  0,  0,  0,  8,  4,  0,0,
450       31, 10,  0,  1,  0,  0,  0,  8,  4,  0,0,
460       31, 10,  0,  1,  0,  0,  0,  2,  4,  0,0}
470   m_vset(74,v)
480   vset2(94)
490   v={               /*      e l e c t r i c  b a s s
500 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
510       58, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
520 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
530       31, 14,  6,  3,  2, 33,  0,  0,  6,  0,0,
540       19, 15,  5,  3,  4, 20,  2,  6,  6,  0,0,
550       31,  4,  5,  3,  1, 40,  0,  0,  6,  0,0,
560       31, 13,  6,  3,  5,  0,  0,  1,  6,  0,0}
570   m_vset(75,v)
580   v={               /*      e l e c t r i c  o r g a n
590 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
600       44, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
610 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
620       31,  0,  0,  0,  0, 22,  0,  2,  7,  0,0,
630       17,  9,  0,  2,  1,  0,  0,  8,  7,  0,0,
640       31,  0,  0,  0,  0, 23,  0,  4,  3,  0,0,
650       17,  9,  0,  2,  1,  0,  0,  4,  3,  0,0}
660   m_vset(76,v)
670   vset2(96)
680   v={                          /*      t u b l a r  b e l l
690 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
700       45, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
710 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
720       19,  9,  0,  4,  1, 33,  0,  1,  4,  0,0,
730       19,  9,  0,  3,  2,  0,  0,  1,  4,  0,0,
740       19,  9,  2,  4,  3,  0,  0,  2,  4,  0,0,
750       19,  9,  0,  3,  2,  0,  0,  2,  4,  0,0}
760   m_vset(77,v)
770   vset2(97)
780           v       ={       /*      s y n t h e  b r a s s
790 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
800       58, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
810 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
820       12,  5,  2,  3,  1, 25,  2,  2,  3,  0,0,
830       14,  7,  0,  3,  1, 32,  1,  6,  5,  0,0,
840       20,  6,  0,  3,  2, 42,  0,  2,  3,  0,0,
850       17,  3,  0,  3,  2,  0,  1,  2,  4,  0,0}
860   m_vset(78,v)
870   vset2(98)
880           v       ={       /*      s y n t h e  b r a s s
2
890 /*    AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
900       58, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
910 /*    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
920       16, 14,  0,  8,  0, 27,  1,  1,  0,  0,0,
930       15, 12,  0, 10, 15, 63,  1,  3,  0,  0,0,
940       20,  0,  0, 10,  0, 47,  0,  1,  0,  0,0,
950       16,  0,  0, 10,  0,  0,  1,  1,  0,  0,1}
```

▶つい先日，「ジェノサイド」を買ったんですが，サウンドの凄いこと，凄いこと。特にZOOMのタイトルの次に出てくる，サンプリングのオーケストラヒット(たぶん)が凄い！ほかのゲームの音楽が，チャッちく聞こえますよ！　尾松　剛 (16) 京都府

```
 960   m_vset(79,v)
 970   v=(                        /*        h i h a t   c l o s e
 980 /*     AF  OM  WF  SY SP  PMD AMD PMS AMS PAN /
 990        61, 15,  0,  0,  0,  0,  0,  0,  0,3,  0,
1000 /*     AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AM-EN
1010        26, 10,  5, 10, 10,  0,  2, 14,  0,  3,0,
1020        31, 20, 11, 15, 11,127,  1,  3,  7,  1,0,
1030        28, 24, 10, 15, 10,  2,  0,  4,  7,  1,0,
1040        19, 21, 12, 15, 11,  0,  0,  2,  7,  1,0)
1050   m_vset(80,v)
1060 str p1="y3,1",p2="y3,2",p3="y3,3"
1070 /*
1080 str rd="y2,4",cL="y2,8",th="y2,11",tm="y2,12",tL="y2,13"
1090 str bd="y2,2",ms="y2,10",sd="y2,16",scy="y2,5",kc="y2,23"
1100 str bs="y2,9",cy="y2,3",hc="y2,6",ho="y2,7"
1110 str et1="y2,58",et2="y2,59",et3="y2,60",et4="y2,61"
1120 str tb1="y2,32",tb2="y2,33",tb3="y2,34",tb4="y2,35"
1130 /*
1140 key 2,"m_play()"+chr$(13)
1150 /* m_init():m_trk(1,"t140L16@80v8o2q8p3"+c):m_play()
1160 m_init():m_tempo(144)
1170 for i=1 to 8:m_alloc(i,3000):m_assign(i,i):next
1180 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
],z[256]
1190 str aa[256],bb[256],cc[256],dd[256],ee[256],ff[256],gg[256
],hh[256],zz[256]
1200 str b1[256],b2[256],b3[256],b4[256]
1210 str c1[256],c2[256],c3[256],c4[256]
1220 /*
1230 /*
1240 /*         m e l o d y
1250 /*
1260 /*
1270 a="L8d.f.a
1280 b="L8|:g2.def2e4c+4|1d1&d2d.f.a:|d1e1
1290 bb="f+4.f2f2e2&ef4.f2f2r
1300 c="e.f.ga2&aa+<cd>a+2&a+agfe2&efga+a2&agfed2&defgd2&defgd2
&defdd4.c+4efga2&aa+<cd>a+2&a+<cdfe2&efga+a.f.ga4<c4>a+d2efga+.a
.<c>a+2a+.a.<c>a+.<c.de.f.gere4f2>ga+<dfererdc+4.g2..f+16e16f+2&
f+
1310 d="a<def+2&f+gaa+g2&gf+edg4.f+16e16f+1&f+>a<def+2&f+edf+f+
2e2
1320 dd="g4.f2&fe4.f2&fg4.f2&fe2
1330 /*
1340 m_trk(1,"[do]")
1350 m_trk(1,"@76 v12 q8 o3 y48,0"+a)
1360 m_trk(1,"o3"+b)
1370 m_trk(1,"@73 v11    o4 p2"+bb)
1380 m_trk(1,"@74 v9     o3 p3"+c)
1390 m_trk(1,"@77 v10    o3 p1"+d)
1400 m_trk(1,"@73 v12    o4 p1"+dd)
1410 m_trk(1,"[loop]")
1420 /*
1430 /*
1440 /*        m e l o d y    (echo part)
1450 /*
1460 /*
1470 m_trk(2,"@w8")
1480 m_trk(2,"[do]")
1490 m_trk(2,"@96 v9  q8 o3 y49,40"+a)
1500 m_trk(2,"o3"+b)
1510 m_trk(2,"@93 v8     o4        p2"+bb)
1520 m_trk(2,"@94 v7     o3        p3"+c)
1530 m_trk(2,"@97 v8     o3 y49,20p1"+d)
1540 m_trk(2,"@93 v9     o4 y49,40p1"+dd)
1550 m_trk(2,"[loop]")
1560 /*
1570 /*
1580 /*         b a s s
1590 /*
1600 /*
1610 a="@w2
1620 b="L16|:grrggrdr:|crrccr>gr arraar<c+r|:4drrddr>ar<:||:grr
ggrdr:|crrccr>gr arraar<c+r>|:a+rra+a+rf|1r:|a+<|:crrccr|1>gr<:|
e8
1630 b1="L16|:d8.<c+64&d32.rrc>cfrarcdf>a+8<a64&a+32.rdfga>a<c+
ergaec+>a:|" /* あとでまたつかうんです
1640 c="<|:drrddr>a<r:||:grrggrdr:||:crrccr>g<r:|frrffrcrfrrfd8
c8>|:a+rra+a+rfr:||:grrggrdr:|<|:erreer>b<r:|>|:arraarer:|<|:frr
ffrcr:||:grrggrdr:||:arraarer:|
1650 cc="frrffre8 drrddr>ar |:a+rra+a+rfr:|<|:d+rrd+d+r>a+<r:||
:erreer>a+<r:|>|:arraarer:||:a+rra+a+rfr:|<crrccrgr >arraar<c+8|
:3drrddr>a<r:|
1660 d="d<dr>afgef|:drrddr>a<r:||:grrggr|1dr:|>a+8<drrddr>a<rc+
rrc+c+r>a<r|:crrccr|1>g<r:|er|:grrggrdr:||:arraarer:||:3drrddr>a
<r:|d<d>c<c>>a+<a+>a<a|:3drrddr>a<r:|d<dr>afgef
1670 /*
1680 m_trk(3,a)
1690 m_trk(3,"[do]")
1700 m_trk(3,"@75 v12 q7 o3 y50,0"+b)
1710 m_trk(3,b1)
1720 m_trk(3,"v13"+c):m_trk(3,cc)
1730 m_trk(3,d)
1740 m_trk(3,"[loop]")
1750 /*
1760 /*
1770 /*         s u b   m e l o d y
1780 /*
1790 /*
1800 a="L8f.a.<c>
1810 b="L8|:b2.gga+2a4e4|1f1&f2f.a.<c>:|g4.f<fga+<d>>g4.f<fa+<c
1820 bb="g>a4.a2g2e2&ea4.a2g2&gg.a.a+
1830 c="d4f4.fa4d4g<d>d>a+ga+<c4g4cega+f4e4cec>aa+4<f4.ga+<d>a+
>a+<f4.ga+<d>a+>a+<g+4<g+4>a+g+g4.g4gaa+c4f4.fa4d4g<d>fd>a+4<c+4
gac+ec+>a<f.>a.<cf4a4gf4>a+4.<d4g.f.d+g2g.f.eg.g.a+<c+.c+.c+c+rc
+4d>a+gfdfga+a+rgraa4.a2d4.d16c+16d2&d
1840 d="ef+aa4<c+d4dde>a+4<de4>a+a+a+a4.<d16>a16<c+4>a4a4.g4ef+
gb4<c+d4>bbb<c+2c+2|:3d4.d2&d:|d2
1850 /*
1860 m_trk(4,"[do]")
1870 m_trk(4,"@78 v10 q8 o3 y51,0 p3"+a)
1880 m_trk(4,b+bb)
1890 m_trk(4,"    v11 o3"+c)
1900 m_trk(4,"    v10 o3"+d)
1910 m_trk(4,"[loop]")
1920 /*
1930 /*
1940 /*         s u b   m e l o d y    (echo part)
1950 /*
1960 /*
1970 a="@w8"+a
1980 bb=">d4.d2d2c+2&c+d4.d2d2&dg.a.a+
1990 c="L16c>afdfa<dfd>afdfa<df|:gd>a+ga+<dga+:||:gec>a+ga+<ce:
|fec>afa<cefdc>afa<cd|:fd>a+gfga+<d:||:gd>a+gfga+<d:||:g+e>bg+eg
+b<e:||:d>aec+dea<c+:||:fc>afa<cfa:||:gd>a+gfga+<d:||:ec+>aec+ea
<c+:|
2000 cc="fc>afcfa<cfd>afcfa<d|:fd>a+gdfa+<d:||:d+>a+gd+>a+<d+ga
+<:||:e>a+geceg<c:||:ec+>aec+ea<c+:||:d>a+gfdfga+<:|c>a+gecega+<
c+>agec+ea<c+d2a4.a16a16a2&a4
2010 d="L8ef+aa4<c+d4dde>a+4<de4>a+a+a+a4.a<c+4>a4a4.g4ef+gb4<c
+d4>bb
2020 dd="g2g2|:3a4.a2&a:|a2
2030 /*
2040 m_trk(5,"[do]")
2050 m_trk(5,"@98 v8  q8 o3 y52,20 p3"+a)
2060 m_trk(5,b+"v9"+bb)
2070 m_trk(5,"@79 v10    o5 y52,0"+c):m_trk(5,cc)
2080 m_trk(5,"@98 v8     o3 y52,20"+d):m_trk(5,"v10"+dd)
2090 m_trk(5,"[loop]")
2100 /*
2110 /*
2120 /*         c h o r d
2130 /*
2140 /*
2150 a="@w2
2160 b="L1gfd2>a+2g2a+4<d4gaa+<c"
2170 bb="d1&d1&d1&d1
2180 c="L8a.a.a2r|:a+.a+.a+2r:|a.a.a2rf.f.f2r|:a+.a+.a+2r:|a.a.
a2ra.a.a2ra+.a+.a+2r<c+.c+.c+2re.e.d2r>|:a+.a+.a+2r:|g.g.g2ra.a.
a2ra+.a+.a+2rg.g.g2r|:a.a.a2r:|
2190 d="d1d+1d1g2.a4b1<c1|:3d1&:|d1
2200 /*
2210 m_trk(6,a)
2220 m_trk(6,"[do]")
2230 m_trk(6,"@71 v12 q8 o3 y53,0 p1"+b)
2240 m_trk(6,"@91 v10              p1"+bb)
2250 m_trk(6,"@72 v11    o2        p1"+c)
2260 m_trk(6,"@71 v11    o4        p2"+d)
2270 m_trk(6,"[loop]")
2280 /*
2290 /*
2300 /*         c h o r d      (echo part)
2310 /*
2320 /*
2330 a="@w2
2340 b="g4&"+b+"2."
2350 c="L8f.f.f2r|:g.g.g2r:|f.f.f2r|:d.d.d2r:|g.g.g2re.e.e2rf.f
.f2rg.g.g2r|:a.a.a2r:|f.f.f2rd+.d+.d+2r|:e.e.e2r:|f.f.f2re.e.e2r
|:d.d.d2r:|
2360 d="d4&d1d+1d1g2.a4b1<c1|:3d1&:|d2.
2370 /*
2380 m_trk(7,a)
2390 m_trk(7,"[do]")
2400 m_trk(7,"@91 v9  q8 o3 y54,40 p1"+b)
2410 m_trk(7,"@75 v11    o3        p1"+b1)
2420 m_trk(7,"@72 v11    o2 y54,0  p1"+c)
2430 m_trk(7,"@91 v8     o4 y54,40 p2"+d)
2440 m_trk(7,"[loop]")
2450 /*
2460 /*
2470 /*         d r u m s
2480 /*
2490 /*
2500 a="@w2L16"
2510 b1=p3+cy+"crcc"+sd+"C8cc|:6"+bd+"crcc"+sd+"C8cc:|"+et1+"C"
+et1+"C"+et2+"Cr"+et3+"Cr"+ cL+"c"+ cL+"c|:7"+bd+"crcc"+sd+"C8cc
:|"
2520 b2=bd+"C"+cy+"c8."+sd+"CC"+sd+"cc"+bd+"ccc"+ho+"C&"+sd+"Cc
cc"+bd+"cc"+bd+"C8"+p2+tm+"c32&"+tm+"c32"+p3+tL+"c"+p1+tL+"C&"+t
L+"C"+p3+bd+"ccc"+ho+"C&"+sd+"Cccc"+bd+"cc"+ho+"C8"+sd+"cc"+cL+"
C&"+cL+"C"
2530 b3=bd+"ccc"+ho+"C&"+sd+"Cccc"+bd+"cc"+bd+"C8"+p2+tm+"c32&"
+tm+"c32"+p3+tL+"c"+p1+tL+"C&"+tL+"C"+p3+bd+"ccc"+ho+"C&"+scy+"C
ccc"+bd+"cc"+ho+"C8"+sd+"c"+sd+"c"+sd+"C&"+sd+"C"
2540 c1="|:7"+bd+"crcc"+sd+"C8cc"+bd+"cr"+bd+"cc"+sd+"C8cc:|"+b
d+"crcc"+sd+"C8cc"+p2+th+"C"+th+"C"+p3+tm+"C32&"+tm+"C32"+tm+"C"
+tm+"C&"+tm+"C"+p1+tL+"C"+tL+"C"+p3+cy+"crcc"+sd+"C8cc"
2550 c2=bd+"cr"+bd+"cc"+sd+"C8cc|:6"+bd+"crcc"+sd+"C8cc"+bd+"cr
"+bd+"cc"+sd+"C8cc:|"+bd+"crcc"+sd+"C8cc"+bd+"cc"+bd+"C"+ho+"C"+
sd+"C"+ho+"C"+ho+"C"+ho+"C"+bd+"crcc"+sd+"C8cc"+bd+"cr"+bd+"cc"+
sd+"c8cc"
2560 c3=bd+"crcc"+sd+"C8cc"+bd+"cc"+sd+"C"+sd+"C"+sd+"Cr"+sd+"C
"+sd+"C"+bd+"crcc"+sd+"C8cc"+bd+"cr"+bd+"cc"+sd+"C8cc"+bd+"cc"+h
o+"C8"+sd+"cc"+ho+"C8"+bd+"cc"+bd+"C8"+sd+"c"+ho+"C"+ho+"Cc"
2570 d="|:9"+bd+"crcc"+sd+"C8cc"+bd+"cr"+bd+"cc"+sd+"C8cc:|"
2580 dd=et1+"c"+et1+"r"+et1+"c"+et1+"c"+et2+"C&"+et2+"C"+et2+"c
"+et2+"c"+et3+"C&"+et3+"C"+et3+"C&"+et3+"C&"+et4+"C&"+et4+"C"+et
4+"C&"+et4+"C"
2590 m_trk(8,"@80 v10 o2 q8 y55,0 p3"+a)
2600 m_trk(8,"[do]")
2610 m_trk(8,b1):m_trk(8,b2):m_trk(8,b3)
2620 m_trk(8,c1):m_trk(8,c2):m_trk(8,c3)
2630 m_trk(8,d):m_trk(8,dd)
2640 m_trk(8,"[loop]")
2650 m_play():end
2660 func vset2(a)
2670 v(0,2)=2     /*      wf
2680 v(0,4)=200   /*      spd
2690 v(0,5)=50    /*      pmd
2700 v(0,7)=3     /*      pms
2710 m_vset(a,v)
2720 endfunc
```

▶ジェノサイドの自機が，まねき猫のバケモノに見えたのは，僕だけでしょうか……。

八尾　唯仁（13）神奈川県

計算機科学者は夢を語り続ける

深夜の議論

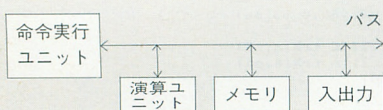
僕は計算機，あるいは人工知能に対して大きな思い入れ，俗な言葉を使うならば夢をもっています。ですから，たとえば「計算機対人間」という対立図式から抜けきれない人や，人間の心は神聖だから計算機でシミュレートするなんてとんでもない（案外そのとおりかも）のだ，計算機は指示されたとおりのことを順番にやるだけだと思っている人（実はこれもいえてるかも）などが近くにいて，しかも話し合う余地がその人にある場合は，無駄になるかもしれない努力をついしてしまいます。

先日も，相手は20歳も目上の分野の違う先生だったのですが，強力な援軍がひとり隣に座っていましたので，深夜2時までその人を相手になんだかんだと偉そうにしゃべってしまいました。

「人工知能なんてとんでもない，心なんて気軽に使ってもらっては困る，僕の心がわかるのか？」

こういう発言に対して，どう答えたらいちばん説得力があるのでしょうか？　僕自身，人工知能は専門でないこともあり，また勉強不足なので，相手の人を納得させるだけの反論をすることは結局できませんでした。確かに心という領域に至っては，人文科学の学問においても，簡単に解決することのできない永遠のテーマですから，その発言に対して真正面から答えることはできなかったのです。

しかしここで，人工知能というものは，人間の内面をとらえるための情報工学的な手法，アプローチなのだという主張をしたらわかってもらえそうな気がします。要するに，人間の知の領域を情報工学的なアプローチでモデル化するというものなのでしょう。したがって，シンボルを用いて論理学的な面から追求する立場や，もっと低レベルでかつ高並列な神経回路網理論的な立場もそれぞれに最終的には同じ人工知能を目指しているという見方も，そう見当はずれということはないでしょう。

図1　究極のRISCアーキテクチャ

演算ユニットもメモリ空間にマッピングされており，特定の番地にデータを書き込むと演算が実行されて特定の番地（出力レジスタ）に出力される。

いずれにせよ，人工知能とか知能機械の研究の成果として出てきたものがきわめて単純なモデルだとしても，それはあくまでも知能の一部をモデル化してシミュレートしているのであって，これこそ人工知能そのものだといっているのではないということに注意しなくてはなりません。手法は違っていても，人文科学の目指しているものと究極的には同じなのです。ですから，

「人工知能，人工知能というが，こういうときに，こうする，ああいうときにああする，というように，パターン化できなくては，実現することは難しい，しかし，パターン化したものはすでに知能といえるものではない」

という世の中でたびたび耳にする批判はピントが外れているといわざるを得ません。

僕が話題にしていること

人工知能の強力な弁護をしていても，自分が人工知能そのものを深く研究しているわけでないので，むずがゆいものがあります。所属している研究室では，「究極のRISCなどと名付けられた記事をもう読んだ?」などという会話は飛び交いますが，「富士通と東北大が右脳コンピュータなるものを作ったって書いてあるよ！」などという発言はあまり聞きません。まあ，部外者であるからこそ，夢のようなことを話していられるのでしょうか（いや違うのだ）。

今回の話と関係はあまりないのですが，この究極のRISCとやらについて，面白いので簡単に触れておきましょう。

RISCといえば人工知能もへったくれもない泥臭い話になりますが，このところ流行しているアーキテクチャのことです。要するに，きわめて簡単な命令だけしか用意しない計算機の総称でして，大学院生数人が数カ月で設計できてしまうということも，初期のころのセールスポイントでした。とにかく今は猫も杓子もRISCです。

「究極のRISC」というやつはアメリカの学会が出している雑誌に載っているもので，僕が以前この連載で究極アーキテクチャシリーズとして冗談半分に書いたものと基本的には違いはありません。

このRISCではついに命令はデータ移動のためのMoveだけとなってしまいました。繰り返しますが，命令はたったの1種類しかないのです。したがって，オペコードを書く必要はなく，ただ単に「移動元，移動先」を書き並べたものがプログラムなのです。

なぜこのような計算機が作れるのでしょうか？　それは，この計算機が「memory-mapped everything computer（すべてメモリマップする計算機）」だからなのです。図1に示すように，実行ユニットから出ているバスにメモリだけでなく，演算ユニット，入出力機器，そして，プログラムカウンタまでがぶら下がっている構成をとっているわけです。そして，それらが特定のアドレス空間を占めるようにマッピングされているのです。

ところで，この計算機のどこに意識があるのかといいますと，

1)　パイプライン化によりそこそこの性能が出せる。

2)　用途に合わせて自由に拡張できる。その際，自然に命令が強化されることになる。

ということが挙げられています。市販のシークエンサを使って簡単に作ってみようという気になり，拡張もいろいろ自由にできるということが，まあ最大のセールスポイントといえるでしょう。

「スーパーストリーム」批判

人工知能や計算機にそこそこ興味がありながら，結局はそれに対する理解が足らず逆に人工知能に対する批判に陥るケースは案外多く，しかも手強いものがあります。最近読んだ本，『知能のスーパーストリーム』について，ちょっと批判的に紹介してみることにしましょう。

この本では，多くの実験例を紹介しながら，動物から類人猿，人間へという連続的な「知」の流れの存在を述べています。知能の持つ，進化における連続的な側面については僕自身も考えるところがあり，たいへん興味深く読みました。

たとえば，動物の持つ表象能力や前言語

能力の存在についての多くの実験は，知というものが人間だけが持つものではないのではないかという疑問をリアルに浮かび上がらせてくれます。

本書の後半ではいよいよ並列計算機，人工知能，神経回路網理論に対する批判が始まります。そこで基本的に間違っていると思うのは，現在多くの研究が活発に行われており，日々多少なりとも成果を上げ，わずかではあるが人工知能というものに近づきつつあることを考慮せずに，現在における人工知能の姿を固定したものと捉えて批判している点です。以下にいくつかの象徴的な部分を挙げます。カッコ内は僕のコメントです。

●並列処理コンピュータの並列処理の中に人間特有の知的活動の生起を期待することは，（中略）フラスコに原始大気をつめて，火花放電を浴びせることによって，フランケンシュタインを作り出すことを期待するようなものといってもよいだろう。

（並列処理は近い将来的には人工知能の必要条件となると思う）

●神経回路網がいかに脳にヒントを得ていようと，それが脳の神経回路をモデル化していることにはならない。

（モデル化の出発点はヒントにすることから始めるのでは？）

●つまり，人工知能とはコンピュータでは実現不可能な概念なのである。

（そういう定義をしたいのならば自由にしてください。ただし，一般の定義とは異なるのでは？）

●コンピュータは人間に近づけば近づくほど，人間にとって役に立たないものになっていく。（中略）進むべき方向は逆である。コンピュータは人間に近づいてくるべきものではなく，人間を拡張するものである。

（逆の方向とは人間が超能力を身につけてコンピュータに近づくことでは？人間を拡張するためにコンピュータを近づけているのです）

筆者は，人工的に作り上げる目標とする知能というものを，（本書の前半では，動物との連続性の一面を指摘したのに）とてつもなく，崇高なものとし，一方，現在進行形の人工知能については，ごく限定された現在の形をその言葉の定義のようにして，その落差を印象付けているのです。

終わりのほうの章では，現在の人工知能をけなすだけけなしたので，ついに「スーパーI」なるものについての提案がなされています。

「私たちはこれから先，コンピュータと共に未来に向けて歩み，新しい次元での動的発展を遂げていくことになる。私たちと共に歩むこの種のコンピュータは人工知能ではない。私はこれをスーパーIコンピュータと呼ぶ」

この文章の書いてある部分の4ページ前の「科学と技術は，（中略），コンピュータと同様，人の後にあるべきものである」という表現と整合しないなどということは小さい問題です。人工知能について批判するだけして，もっと素晴らしい何を提案しようとしているのでしょう？

最終章の「スーパーI，超知能へ」で，従来の人工知能との違いについては，人の知的能力と異なる知的能力を発揮することによって人と共進化するということを挙げています。しかし，いったい，人間の持たない知的能力とはいったい何なのでしょうか？（まさか超能力ではないでしょう？）いずれにせよ，最終章でも，スーパーIが何たるかはほとんど示されず，自然言語理解や曖昧な判断などは無益だなどと決めつけています。

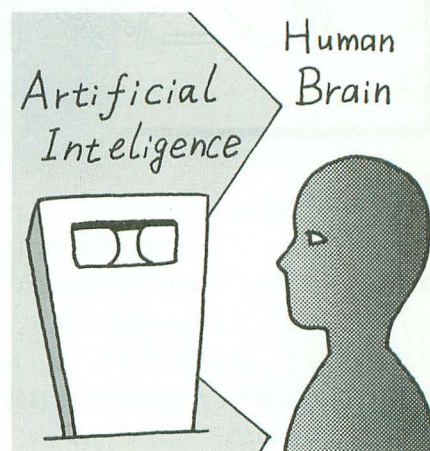
計算機や人工知能研究の表面的な部分しか見ずに外側から攻撃をし，しかも中身のない代案をされるのはいやな感じです。そしてはっきりこういうのです。

「科学や技術に携わる人々は，夢の語り部であってはならない」

僕は「不良技術者」ということになりますね。

計算機に対する思い

さてここで，Macを生み，次にはNeXTをもたらした，あの夢見るスティーブン・ジョブスの日本での製品発表会の模様について，書いてみることにしましょう。

会場ではアメリカでの発表会と同じように生バイオリンとNeXTの共演が再現され，NeXTについてのジョブスの講演も行われました。彼はそこで相変わらずキャッチーな（この言葉を1度使ってみたかった，でももう2度と使わん）ことをいっています。

要するに，これからは計算機は芸術の分野にどう入っていくかが問題だということです。まったく我が意を得たりという感じです。すでに計算機は合理化のための道具や暇を作り出すだけではなくなりつつあります。そして，合理化のために生まれた時間をより有意義にするものになりつつあると思っています。これは，人類の歴史にとって大きな転機であると考えられます。

日本でも，たとえばTRONのように長期的あるいは広域的な視点を持ったプロジェクトが起こってきています。その技術的な側面は横に置いておいても，トップダウンにものを作っていくという環境が日本にもできつつあることは望ましいことでしょう。ただ単に速いというだけでなく，未来において誰がどのように使うかを，どの階層を担当している技術者も，頭の片隅にはいつも置いていられたらなぁと思います。

そして僕も，「究極のRISCってちょっとリスクが大きい」だとか，「ノイマンは肥満体だ 」などといいながらも，行き帰りの電車の中などで，これからの人間とお茶目な計算機たちとの付き合いについて，ぼんやりと思いを巡らしているわけなのです。

参考文献

1) D. W. Jones, "The Ultimate RISC", Computer Architecture News, Vol. 16, No. 3, pp. 48-35(1988).
2) G. W. Griffin, "The Ultimate Ultimate RISC", Computer Architecture News, Vol. 16, No. 5, pp. 26-31(1988).
3) 半田智久，「知能のスーパーストリーム」，新曜社（1989）。
4) 曾和将容，有田隆也，「CP コンピュータの提案」，電子情報通信学会研究報告，Vol. 89, No. 55, pp. 1-7(1989)。

第39回

猫とコンピュータ
ホンニャア・IN・テクノ書斎

Takazawa Kyoko
高沢 恭子

生まれつき毛皮を着てるってことは夏にはかなりのハンディじゃないかなあ，と動物ごとながら気になります。暑さのせいか少々活気のないホンニャアを横目に，キョウコさんはZ'sSTAFFを試してみました。

ホンニャアはだまっているけれど，これはやっぱり暑い。でもこの家のリビングは衛生第一というのでじゅうたんを敷かないから，夏こそまったくありがたいと思う。木目の床の上を，あっちにゴロリ，こっちにゴロリ，ひんやりした場所を選んで移動していけばよいのだ。

さっきから風も息を止めた。お日さまも芝生をジリジリ熱しながらだんまりを続けている。だあれもやって来ないし，なんだか耳なりが聞こえそうに静かだ。ホンニャアはまたゴロリと寝返った。

オトコ一匹

こんなとき，犬じゃなくてほんとによかったなと彼は思う。アイハラさんちのハチなんか鎖につながれて庭暮しだから，真夏の午後は毎日ホットドッグになる。同じ毛皮を着せられていても，猫はあんなにダラリとベロを出さずに夏を過ごせるのは，体の大きさもちがうけれど，涼しい場所をみつけて自由に動きまわれるからということもあるだろう。

つながれたり，閉じ込められたりして飼われている犬に比べたら，猫というのはまったく天下御免の奔放さだ。そのために都合よく暮らせるし，カッコよく生きられるなと自分でも思う。親も兄弟も捨て，裸一貫，無一文だ。まあ，飼い猫なので大きいことは言えないが，オトコ一匹の気分で生きているつもりだ。

こんなふうに，暑くてアメみたいにノビきっているとサマにならないが，明け方の公園や月影の塀の上なんかを歩いているとけっこう孤独を愛している感じがするもんだ。寒いときのほうがもっとカッコイイ。北風の中をさっそうと，石垣から屋根伝いにとび歩く様子は，まさに一匹オオカミの快感だ。

そこへいくと，同じ1匹でも犬は寂しそうでいけない。本人は意外に明るくひょうきんでも，夕暮れや寒風の中の犬のひとり歩きはなんだかあわれに見えるからふしぎだ。犬はつながれて，人といっしょに賑やかに暮らすのが似合っている。猫はうまいぐあいに人に可愛がられたりもするが，気ままなひとりがなぜか似合うのだ。

それにしても暑くてまいった。ホンニャアはまたゴロリと寝返ると，体が庭のほうを向いた。そのとき，背の高いムクゲの幹の根本が動いて誰かの影があらわれた。とがった大きめの耳と長いシッポ，右目のまわりだけまっくろの白黒猫，独眼竜のジャブだった。相棒のフックの姿はない。

「なんだ，おマエか……」とホンニャアはつまらなそうに，一度起こしかけた体をまたゴロンと横たえた。どうもあいつはケンカ好きで困る。遊んでいてもかならず最後はワケのわからないケンカになる。

ジャブは網戸のそばまで近づいてきて部屋の中のホンニャアをのぞきこむようにした。遊ぼうよというのか，昼間から猫の集会があるので呼びにきたのか。

「やだよ，第一暑くてたまらん」というふうにホンニャアはそっぽを向いた。おマエとのつきあいは刺激的すぎて夏向きじゃないんだ。ジャブはあきらめきれずにじっと室内のホンニャアをみつめていたが，やがて暑さでキゲンの悪いハチに吠えたてられて，焼けつく芝生の上を走り去った。

電子アトリエにて

ホンニャアは起き上がってマシンルームのドアの前までいくと，「ニャア」とひと声ないた。これで中から扉が開いて，部屋に入れてもらえる。だいたい午後の2時を過ぎるとマシンルームはクーラーのスイッチが入れられて，トオルか私がパソコンをいじったり本を読んだりしている。ホンニャアはそれを知っていて，時々昼寝にやってくるのだ。クーラーは性にあわないのだけれど，毛皮の暑さに耐えきれないときはやむを得ない。

マシンルームではトオルが，私にZ'sSTAFF PRO-68Kの技法のあれこれを手ほどきしてくれるのだと言って，たいへんな素早さでいろいろな絵を描いていた。

「ねっ，グラデーションの中にさらに細かいグラデーションをつくることもできるでしょ」

太めのペン先を選んで，高層ビル群を俯瞰のアングルからとらえた構図を描き，そのアウトラインの中をさまざまに塗りかえて見せている。

ビルの壁面が変わる。空が変わる。そのたびに，あかね色の空にそそり立つ勇壮なビルになったり，ライラック色の空に濃紺の影を見せる夕闇のビルになったりする。空とビルの色がもっと似通って青みが増したら，深海の底のガラスの墓標に見えたりもした。

「タイルやってみるね」

パレットのウィンドウの中からさらにタイルのウィンドウを開き，ひとつのパターンをセレクトする。ビルの右がわの壁はスルスルと一面サクランボの柄に塗られた。

「こっちがわはお星さまにしよう」

左の壁は星の図柄，空は千代紙みたいなパターンになった。トオルは「この上にも別の柄を描けるよ」と，どんどん違うパターンを塗りかさねていく。つぎはエアーブラシ，それからマスキング，そしてトランスフォーム。画面はあらゆる部分が手あたりしだい加工されていくので，はじめは具象画ふうであったものが，抽象画のかたちになってきた。

私もまねていろいろやってみると，画面

は当然，いじるたびいくらでも変わる。命令がイタズラであろうとミスであろうと，なんの迷いもみせずにいさぎよく変貌していく。

偶然につくられたコンポジションが，発色自体の美しさや着色の一様性で，じゅうぶん絵画になってしまう。創意なんかなくたってけっこうイケるじゃないか。なにかバカバカしい気もしてくる。

これはよほど技法を熟知した上で，冷静に作業をしなければ，パソコンのワザに圧倒される。あるひとつの色が，その色単独で強力な発言をする。そんな色を絵の具でつくり出すにはたいへんな技術と時間のかかることなのに，パソコンはそれを数知れず用意して待っているのだ。こちらが選ぶつもりが，ついゆきずりの色の鮮やかさにとらわれたりもするだろう。

豊富すぎるほどの編集の機能も，あそび半分では収拾のつかないものになりかねない。目先の変化の面白さに負けて，パソコンに支配された絵をつくるのはミジメだ。

男はねむいよ

ホンニャアは室内を見回して，寝心地のよさそうな場所をさがしている。だいたいこの部屋では，本棚のてっぺんやマシンを納めたラックの頂上に登ることが多い。でもきょうは98のキーボードに決めたようだ。

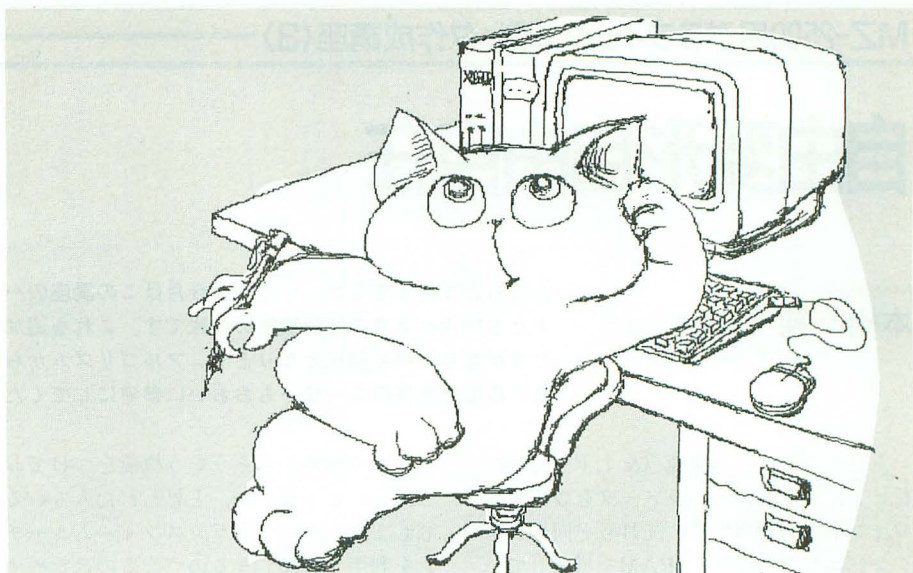
キーボードなら最近は流線型のX68000が気に入っているのだが，きょうは私とトオルがそばにいてぐあいが悪い。それで左隣のデスクにあった98にしたのだ。キーボードは子猫のときのように体まるごと乗せるのは無理なので，このごろは枕がわりにして上体だけを乗せる。

ホンニャアはまずキーボードに片ひじをついたかたちで寝そべり，胸とおなかをひろげ，自分でのぞきこむようにしながらペロペロなめはじめた。これから本格的に寝入る準備だろうか。

「ホンニャア！」

そのリラックスしたようすに誘われて，私は思わず声をかけた。

「ん？　なんだい」

なめるのをやめて首だけこちらに向けたホンニャアの顔は，落ち着きとゆとりがあってまるで人間のようだ。どこかに知性をかくしているんじゃないかな。

「そうだ，ホンニャアを描こう。モノクロームで線描だけで」

「寝てるところ？」トオルが聞いた。

「パソコンのプログラムのアイデアを考えているところ，X68000をバックにして」

「ホンニャア・IN・テクノ書斎だね」

クロッキーブックに，鉛筆でかんたんな構図をつくった。デスクに向かっていたホンニャアが，椅子を回転させてこちら向きになり，もっともらしい顔でプログラムの構想を練っているところだ。デスクにはX68000らしいパソコンが乗っている。

これをマウスで描いてみよう。まずいちばん素朴なことをしてみるには線描だ。鉛筆の味が出るだろうか。パソコンの画面の中でこしらえる絵は，現実に描く絵とはまったく別の分野と考えたいので，手描きと同じことができる必要はないとは思う。でもあえて鉛筆やコンテの感覚で描いてみよう。それはそれで新しい味が出るかもしれない。

ペンのウィンドウを開き，ケシゴムで画面を一掃した。まずはじめに細めのペンを選んで思いきって線を入れてみたが，こまかい動きを出すにはまだ太すぎる。そこでいちばん細いロットリングペンに変えてみた。ちょっとデリケートすぎるけれど，荒っぽくなくていい。

それですぐさま鉛筆の感覚で描こうとしたのだけれど，そうはいかなかった。マウスは鉛筆ほど巧みなカーブはできないし，いきなり意のままに動かすというのもたやすくない。手つきも違う。それなのに鉛筆のときのようなスピード感で，一気にしあげようとして何回もしくじった。

そうか，何もすばやく描こうとしなくていいのだ。選んだペン先の太さは一定なのだし，鉛筆のような濃淡や速度感はもともと無理なのだった。それならゆっくりやってみよう。

そこでゆっくりゆっくり，描いてみた。机やパソコンの線も直線の描法を使わずに描いた。失敗すると，太いペン先に変えてパレットの中から白を選んで塗り，修正した。ペンとパレットのウィンドウを，交互に閉じたり開いたり。そのくりかえしで，だんだんマウスをクリックするテクニックやタイミングがじょうずになった。ただし絶え間なく行われるウィンドウ操作は，どうしてもなかなかの煩わしさである。パソコンで絵を描こうとするなら，これをまず乗り切らなくてはならないのだろう。

かんたんなものだが，「ホンニャア・IN・テクノ書斎」はできあがった。

「できたできた！」と私。

「プリントしてみようよ」とトオルは印刷のウィンドウを開いた。サイズ，範囲，濃度，プリンタのタイプの指定がある。描くことも編集も印刷も，画面の中でマウスで処理されていく。このあたりはやっぱり便利で楽しいなと思う。

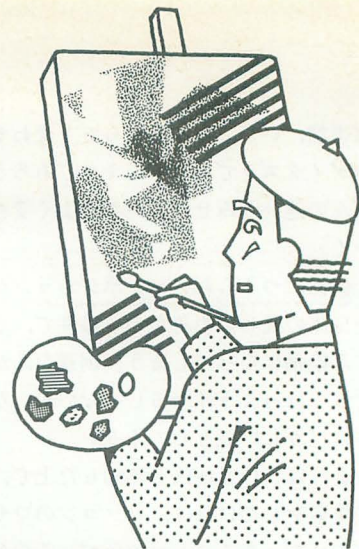
「わぁ，きれいにできたね！」

トオルの声に，寝ていたホンニャアは目をあけてこちらを見ると，

「なんだ，ヘタクソ」という顔をして，また眠りに入った。

自由変形自由自在

Motohashi Jun
本橋　純

お待ちどうさまでした。いよいよ今月はこの講座のハイライトともいうべき自由変形機能の解説です。これを追加することでかなりの機能強化となります。アルゴリズムから解説しますので他機種のユーザーもおおいに参考にしてください。

トランスフォーム機能（&上下・左右反転）と，文字描画のパラメータを設定するウィンドウの登場です。先月号と同様にマシン語エリアを確保し（RAMを増設していれば，CLEAR &H4000），

BLOAD "画餅AMA-25h", ADR

で読み込みます（ADRは入力したときの先頭番地）。あとはMACINTO-Cなどのマシン語入力ツールを使ってリスト1～3を打ち込んでいきます。

なお，上下・左右反転が独立したリストになっていますが，トランスフォームと一緒に入力してください。両者のうちでメインとなるのは当然前者です。とはいっても，トランスフォーム自体の機能としての利用よりも，拡大・縮小（ときには回転）に利用するほうが多いと思います。

操作方法

1)　トランスフォーム機能

左ボタンでまず矩形の範囲を指定します。すると図1のような表示がされます。図を見て察しのよい人はもう変形のやり方はわかったと思います。5つの小さな四角が中心と四隅にありますね。ここで左クリックすると四隅の場合はその隅が移動できるようになりますし，中心の位置では範囲全体が移動可能になります。で，適当な位置に動かしたなら左クリックして位置を固定します。そんなこんなで，移動，変形ともに指定が終了したら，5つの四角のどこでもいいですからその内部で左ダブルクリックを行います。それで描画が開始されます。

あと，上下・左右反転は左ボタンで矩形を指定すればよいだけですので問題ないでしょう。

拡大縮小や回転はトランスフォーム機能で実現できます。とはいってもフリーハンドで四隅をぐりぐり動かして拡大縮小などを行うという結構恐ろしい手段ですけど。

それなら初めからそういう機能をつけておけばいいじゃないか，と思われた人もいるでしょう。すべてトランスフォームルーチンを利用すればできるので作るのはたやすいのですが，それを入れるだけのメモリがもうないのです（残りメモリはあと80Hくらい）。A000H～F77FH がプログラムに割り当ててあります。当初はこれだけあれば十分と思っていたのに。くそー，こんなことならプログラムもバンクチェンジ可能な設計にしておくべきだった。

2)　文字ウィンドウ

文字ウィンドウは単にIOCSのSYMBOLルーチン用のパラメータを設定するだけのつまらないウィンドウなので，さっさと片づけます。

右側に横方向，縦方向の大きさを変えるためのレバーがあります。これはいちばん下が1倍で，上が7倍となっています。そして，8，16といった文字がありますが，これで半角文字のドット数を設定します。あと，文字の方向は4つの矢印が密集したアイコンで指定します。で，矢印が→のときは左から右へ文字を書いていく，ということです。↓のときは文字のフォントが90°回転して上から下へ書いていきます。

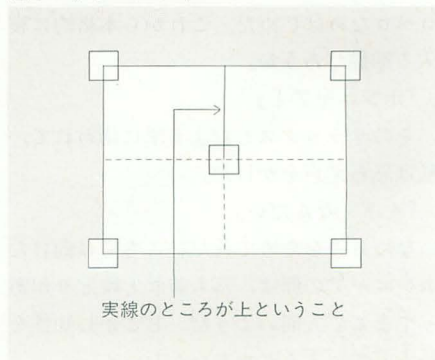
解説：トランスフォーム

解説の前にこのルーチンの制作過程の話をしておきましょう。題して，「トランスフォーム誕生秘話（いま明かされる彼の過去とは？　還らざる時の終わりに彼らはなにを見たのだろうか……）」

そもそも，DMACSでこの機能が動いているのを見たのがこのルーチンとの出会いでした（Z'sSTAFFで見たのではないところが哀愁漂う）。その描画過程を見ているとき，アルゴリズムが思い浮かんできたのです。ラインルーチン（の移動処理）を流用すればできると。

しかし，そのときは頭の中で考えただけで実際に作り始めたのはもっとあとのことでした。なにしろ，その頃は画餅のシステムも完成していなかったのですから。9月になり，画餅の制作はいったん休みにして別の独立したルーチンを作っていました（なにしろ試験期間だったもんで）。そのとき作ったのが，「700に負けるな。テキストVRAM用拡大縮小回転PUTルーチン（笑）」なのでござる。ここで前に考えたアルゴリズムを使ってみたわけです（ああ，ゲームにしか関連づけられない自分が悲しい……）。結局，このルーチンは“アフター某”させるには遅いルーチンでしたが，ここで得たノウハウはトランスフォームルーチンを作る際に役立てられたのです。

このようにゲーム用のPUTルーチンが先祖となっているので，若干精度に問題があって多少汚くなります。けど，DMACSを

図1　トランスフォーム

実線のところが上ということ

320×200ドットで見れば，たいして精度的な違いはなさそうですし。それに参考書というものにまったく依存せず，独自に考えたものだということで勘弁してください。ま，画餅ですからなんでもありですしね。余談ですが，先祖のPUTルーチンはバージョンアップされて，“アフター某”くらいならできそうな速度になっています。

それじゃあ，ぼちぼち解説をしていきましょうか。

このルーチンの描画方法の概要は図2に示すようになっています。元の図形でのデータは上から下へ1ラインごとに位置を移動させます。それに応じて描画側の図形ではデータ側の直線に対応した直線を描画していきます。次からこれらをより詳しくかつボトムアップに述べていきましょう。

0） ラインルーチン

先月のラインルーチンでは2点間の差の絶対値dx，dyで（dx＞dyのときなら），ST＝dy/dxのようにし，dx移動する間にdy移動するようにしていましたね。それを今回トランスフォームルーチンで用いるときは2点間の距離lx（＝dx＋1），ly（＝dy＋1）で，ST＝ly/lxとして処理することにします。lx移動する間にly移動するようになるわけです。これだと1ドット多く描いてしまいそうですが，さにあらず。

1：ドットごとの描画
2：S＝S＋STで桁上がりが起こったらy移動
3：x移動
4：lx回だけ1～3を繰り返す

このようにすれば，結局描画するのはlxドット分になります。さらにSの初期値S0を0≦S≦STの範囲で設定します。これはたとえば，水平方向のラインのときはly＝1でありy方向の移動が起こってしまうような気がします。しかし，前述の範囲でSを設定すればlx回目の描画の際のSは，

$$S=(lx-1)*ST+S0=1-ST+S0<1$$

であり，Sは桁上がりしていなく，y方向の移動が起こりません。このように，Sの初期値の設定をするだけでdx，dyを用いる場合と変わらなくなるわけです。

ではなぜlx，lyを用いるのかというと，私がトランスフォーム処理を考える際に2点間の差で考えずに距離で考えたという単にそれだけのことです。それに，画餅の除算ルーチンは除数に0がきてもエラーにしてないので平気ですが，ほかの処理系で実行するときなどにdx＝dy＝0の場合をチェックするよりはlx，lyを用いたほうが簡単なような気がしますしね。

1） 直線描画

1次元データと描画する直線が図3のようになっている場合を考えていくことにします。このときの処理はラインルーチンの点を打つ部分で描画カラーをデータによって与えるように変更すればできます。データの位置もX0からX1まで移動させていくわけですが，その移動速度は描画する直線の長さに応じて（描画直線の点の移動より）速くしたり遅くしたりすることが必要となります。これを行うためには先月のグラデーションルーチンよろしく，ラインルーチンでの移動方法を流用します。

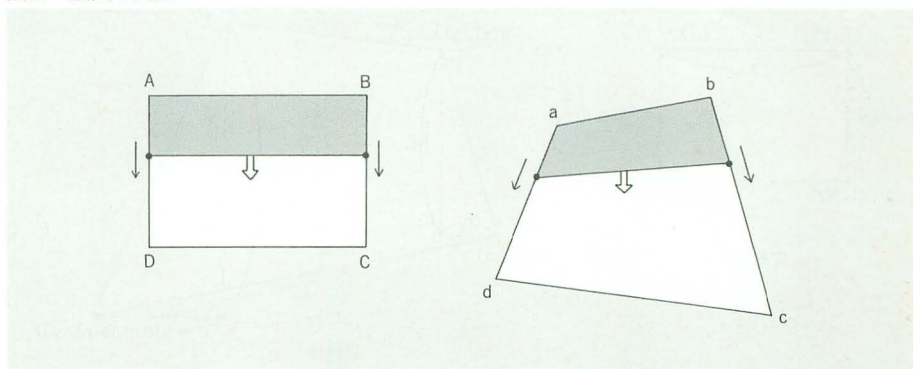
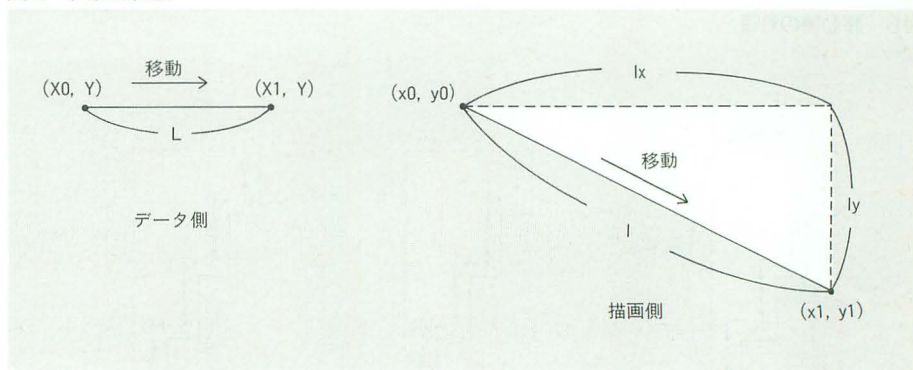
直線の長さはlですから，これとデータの長さLとでデータ位置の移動速度を調整すればよいわけです。しかし，l＝sqrt(lx＊lx＋ly＊ly）であり，lを求める際に乗算2回と√ 1回の計算が必要となります。このことで処理が重くなるのは必至です。そのため，画餅ではlの代わりにlx，lyのうち長いほう（ここではlx）を用いるようにしてしまいます。これが精度を落とす原因のひとつですが，所詮320×200ドットです。正確にやったところで大差ないでしょう。

そうすると，図ではlx＞Lですので移動処理はlx回繰り返されます。描画側もlx＞lyであり同じ回数だけ繰り返されるので，カウンタを共用できますね。もし，共用できないなら，描画側とデータ側の位置の移動が同時終了するよう調整する処理が必要

図2　描画の実際

図3　直線の設定

図4　処理の変更

となります。よって，繰り返し回数は一致したほうが都合がよいわけです。しかし，これはlx>Lの場合の話です。lx<Lの場合は繰り返し回数がLで一致しません。ということは前述の調整処理が必要になるのでしょうか。この問題を画餅ではlx<Lのような場合に限り次のような処理をするようにして解決しています。

繰り返し回数はLでなくlxにしてしまいます。Sd，STdは通常のまま（STd＝10000H＊lx/L；Sd＝Sd−STd)。そして，通常の処理

1：Sd＝Sd＋STdでCYが立ったらその位置のカラーデータで１ドット描画

2：データ位置の移動

を次のように変更します。

1：Sd＝Sd＋STdでCYが立ったら３へ

2：データを移動させ，１へ戻る

3：１ドット描画

つまり，次の桁上がりの位置まで一挙に移動しているわけです。図で示すと図４のようになります。

このようにしてデータ位置の移動処理を作ると，全体の処理は，

ST＝10000H＊ly/lx；S＝ST−1；

1：データ位置の移動と１ドットの描画

2：S＝S＋STでCYが立ったらy移動

3：x移動

4：1〜3をlx回繰り返す

のようになります。

2）　左右のライン

元の図形（矩形）と，変形後の図形の座標や長さが図５のようになっている場合を考えます。概略で述べたように元の図形のデータ位置は上から下へ１ラインごとに移動します。直線ADと直線BC上をそれぞれ(X1，Yi)，(X2，Yi)(Yi＝Y1，Y1＋1，……，Y2)のように移動していくわけです。そして，(X1，Yi）は変形後の図形の直線ad上の点に対応し，(X2，Yi)は直線bc上の点に対応します。そのため，直線ad，bc上を移動する処理が必要です。これは当然ラインルーチンを利用します。この移動処理で注意しなければならない点がひとつあります。

通常の移動は図６の（1）のようにly<lxのときでは，y方向の移動を行う際にx方向の移動も行います。これで移動させて描画を行うと描画図形に隙間ができてしまいます。極端な例では45°の回転をさせたら図７の（1）での斜線部が描画されず，隙間になってしまいます。この問題を解決するには，前述の場合ならy方向の移動をする際に同時にxを移動させないようにします。すると，図７の（2）のように隙間はなくなりますね。この処理をラインルーチンの若干の変更で実現するには図６の（1）の場合では次のようにします。

・ST＝ly/(lx＋ly)；S＝ST−1と設定。すると繰り返し回数はlx＋lyとなる。

・S＝S＋STで桁上がりがあればy移動し，そうでなければx移動。

・データ側移動処理でlx＋lyを描画側の長さとする。

これでよいわけです。この方法で移動させると図６の（2）のようになります。

しかし，画餅ではこの方法は使っていません。データ，描画側の移動とも変更を加えず，距離の短いほうの移動を行うあとに1)の直線描画ルーチンを呼んでしまうのです。つまり，図６の（3）のように，

1：1)を呼び出す

2：S＝S＋STで桁上がりがあればy移動して，1)を呼び出す

3：x移動

のようにします。これでは同じデータで直線を重ねて描画してしまう欠陥があります（精度を落とす原因その２)。しかし，そもそも先祖がゲーム用でしたので，前述の方法を用いたとしてもたいした違いがないのでこれでいいでしょう。

図５　移動の例

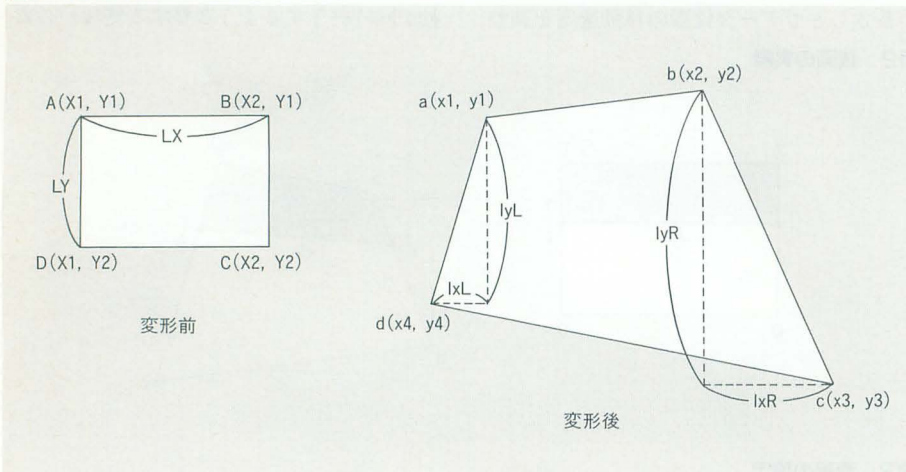



図６　移動時の処理

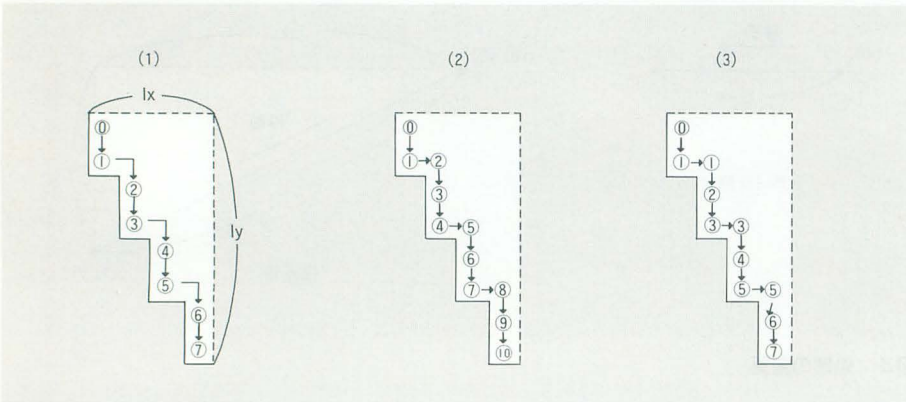



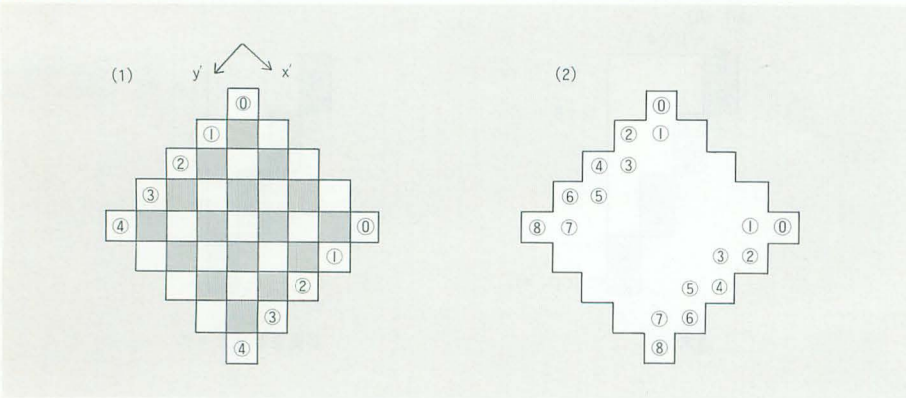
図７　隙間をなくす

3)　左右ラインでの移動

図5での説明に戻りましょう。左右のラインの移動そのものは2)で解説したとおりですが，だからといって両方が同速度で移動してはうまく変形後の図形を描いてくれません。そこで，座標c，dへ同時に到着するようにそれぞれの移動速度を調整する処理が必要になります。

これにもラインルーチンを流用します。長さは1)のときのように直線の長さそのものを用いず，lyL，lyRを用いることにします。この場合は，

STlr＝10000H＊lyL/lyR；Slr＝STlr－1；

1：Slr＝Slr＋STlrでCYフラグが立ったら2)の処理を呼び出して左ラインの移動
2：2)の処理を呼び出して右ラインの移動となります。

4)　y方向データ位置の移動

3)でのlyL，lyRのうち長いほうとデータ側のLYで，データ位置の（y方向の）移動をさせます。これは1)でのデータ移動ルーチンで処理します。うーん，簡単な解説。

5)　まとめ

1)〜4)の処理をまとめて変形処理を行います。これは次のようなかたちになります。

1：1)の処理を行う
2：設定回数だけ繰り返したなら終了
3：3)の処理を行い，その結果として左右どちらかのラインを移動させる。すなわち，2)の処理を呼び出す
4：4)の処理を行う
5：1へ戻る

ここでは，2で終了判定をしています。最後で判定してもよさそうですが，そうすると図5で点cまたはdが画面の端であるような場合に問題が生じます。(繰り返し回数がlyなら) lyR回目のときは1) の描画は直線cdとなります。ここで終了させないと3を行いますね。3では2)の処理を呼んでいますが，2)で解説したように左右ラインの移動の際，1)の処理を呼び出すようなときがあります。つまり，画面外へ移動してから1)の処理が行われてしまう可能性があります。しかし，画餅では画面外での直線の描画をサポートしてません。にもかかわらず，呼び出したら……そう，処理があっちの世界へトリップ（暴走ともいう）してしまいます。そーいうわけで，終了判定が2にあるのです。

プログラムについて

このルーチンは1ドットのデータリードと描画の部分のみが各機種に依存するものですので，他機種ユーザーの人もその部分を変更すれば動くと思います（ただし，320×200ドット用ですけど）。そのため，プログラムの説明もしておきましょう。

まず，データ形式は図8のようになっています。こうして，データ側と描画側の形式を統一しておけばなにかと便利です。次に，このルーチンを呼び出すときはメモリ上のワークに座標を設定しておきます。図5の場合でいえば，

ワーク(ux, uy)←(X1, Y1)

ワーク(dux, duy)←(X2−X1, Y2−Y1)

と設定し，変形後の座標(x1,y1)〜(x4,y4)はそれらと同一のラベル名を入れておきます。そうしてからTRANSFをCALLします。各々のルーチンの説明はソースリスト中にうざったいくらい書いておきましたので，そちらを参照してください。プログラム自体は個人的には非常にわかりやすく書けてあると思うので，マシン語初心者も理解できるでしょう。

解説(おまけ)：文字

文字の描画はIOCSコールを利用していることは以前にも述べました。しかし，こ

図8　データ形式

れだと整数倍の拡大と90°おきの回転しかできないばかりか，単色でしか描けません。画餅ではこれで十分だと思っているので利用しているのですが，もっと凝ったことをやろうと思うと独自にプログラムを作ることになります。そこで必要となるのはフォントデータを読み込むプログラムですね。幸い，これはIOCSコール(SVC 67H）に用意しているので，それを利用するとよいでしょう。これはBASICでの cgpat$ に対応しています。

で，利用方法はDEに文字コード（BASICのマニュアル参照）を入れて呼ぶと，DEにフォントデータの格納アドレスが入って戻ってきます。こうして，データを読めば，あとは変形させせようがグラデーションで描こうがなんでもありですね。

フォントを直接ROMから読み出すこともやってやれないこともありません。しかし，そんなことは“グラフィック”のつかない“エディタ”などで高速にテキスト表示を行う以外は無理して作ることはありませんしね。

＊　＊　＊

次回はエフェクト関係（ソフトフォーカス，色調変換，カラー変換）の詰まったウィンドウとファイル関係のウィンドウが登場します。遠近感のある作品などが描け，なおかつその作品がディスクにセーブ，ロードができるわけですね。いままでctrl-GでBASICを起動していた人，お疲れさまでした。それから，説明不足で申し訳なかったのですが画餅はG-RAMを増設しないと正常に動作しません。一部の方にご迷惑をおかけしたことをお詫びします。では来月。

文字表示の例

リスト1　自由変形ルーチン

```
DA33 CD 05 B0 C8 CD 10 BC CD : B0
DA3B 91 C0 C8 CD 2E A6 38 F7 : E9
DA43 AF 32 AC F9 CD 97 A4 CD : 5B
DA4B CE A9 CD C7 DC CD B7 DA : 45
DA53 CD CD A4 CD D2 A2 CD C0 : 0C
DA5B A4 CA AA DA CD 7C DC 28 : 3F
DA63 F2 3A AD F9 FE 02 28 23 : 1D
DA6B CD BB A4 20 E6 CD D2 A2 : 73
DA73 CD 44 BF 28 09 CD B7 DA : 5F
DA7B CD E8 DB CD B7 DA CD C0 : 7B
DA83 A4 CA AA DA CD BB A4 20 : 3E
DA8B E4 18 C8 AF CD 97 A4 CD : 48
DA93 CE A9 CD B7 DA AF CD 97 : E8
DA9B A4 CD F2 DC 3E 01 32 64 : 14
DAA3 F9 CD C8 A4 C3 A2 C0 AF : 06
DAAB CD 97 A4 CD CE A9 CD B7 : D0
-------------------------------------
SUM: 65 14 C7 97 2A FB 4A 00 BAFD

DAB3 DA C3 A2 C0 AF CD 97 A4 : B6
DABB 21 FF 02 22 C0 0C AF 32 : F1
DAC3 CA 0C 21 55 55 22 CB 0C : 9A
DACB CD AF DB CD DC DA CD 74 : 1B
DAD3 DB CD 08 DB 3E 01 C3 97 : 24
DADB A4 21 69 F9 11 C2 0C 01 : 07
DAE3 08 03 C5 D5 E5 06 00 ED : 7D
DAEB B0 21 C0 0C DF 49 E1 01 : A7
DAF3 04 00 09 D1 C1 10 EB 21 : BB
DAFB 69 F9 01 04 00 ED B0 21 : 25
DB03 C0 0C DF 49 C9 21 CA 0C : B4
DB0B 34 21 71 F9 11 6D F9 D9 : 0F
DB13 21 69 F9 11 75 F9 D5 D9 : B0
DB1B D5 CD 43 DB D1 21 69 F9 : 14
DB23 D9 D1 21 71 F9 D9 CD 43 : 1E
DB2B DB 21 AA AA 22 CB 0C 21 : 6A
-------------------------------------
SUM: D4 DD F7 D7 AF 30 03 39 A57E

DB33 65 F9 11 C6 0C 01 04 00 : 46
DB3B ED B0 21 C0 0C DF 49 C9 : 7B
DB43 CD 5F DB ED 43 C2 0C 22 : 27
DB4B C4 0C D9 CD 5F DB ED 43 : E0
DB53 C6 0C 22 C8 0C D9 21 C0 : 82
DB5B 0C DF 49 C9 1A 86 4F 23 : 0F
DB63 13 1A 8E 47 23 13 CB 28 : 2B
DB6B CB 19 1A 86 1F 6F 26 00 : 38
DB73 C9 21 65 F9 01 04 05 C5 : 17
DB7B 06 00 D9 DD 21 C0 0C 06 : AF
DB83 02 D9 5E 23 56 23 EB B7 : 77
DB8B ED 42 DD 75 02 DD 74 03 : D7
DB93 09 09 DD 75 06 DD 74 07 : C2
DB9B EB DD 23 DD 23 D9 10 E1 : B5
DBA3 D9 E5 21 C0 0C DF 4B E1 : B6
DBAB C1 10 CC C9 21 69 F9 11 : FA
-------------------------------------
SUM: DF 49 5F E7 F2 20 DF 98 3BBC

DBB3 00 00 42 4B D9 06 04 D9 : 49
DBBB 7B 86 5F 23 7A 8E 57 23 : 05
DBC3 79 86 4F 23 78 8E 47 23 : E1
DBCB D9 10 EC D9 CB 3A CB 1B : 99
DBD3 CB 3A CB 1B CB 38 CB 19 : D2
DBDB CB 38 CB 19 ED 53 65 F9 : 85
DBE3 ED 43 67 F9 C9 3A FD FA : 8A
DBEB 3D 28 14 87 87 5F 16 00 : FC
DBF3 21 65 F9 19 EB 21 A0 F9 : 3D
DBFB 01 04 00 ED B0 37 C9 D9 : 7B
DC03 2A A0 F9 ED 5B A6 F9 B7 : 61
DC0B ED 52 E5 2A A2 F9 ED 4B : 21
DC13 A8 F9 B7 ED 42 EB 01 C8 : 3B
DC1B 00 D9 D1 DD 21 69 F9 06 : 10
DC23 04 C5 01 40 01 CD 45 DC : F9
DC2B D9 CD 45 DC D9 C1 10 F1 : 62
-------------------------------------
SUM: 4B B8 92 21 73 59 4E B5 E0F6

DC33 DD 21 69 F9 06 04 CD 6A : A1
DC3B DC D9 CD 6A DC D9 10 F6 : A7
DC43 37 C9 DD 6E 00 DD 66 01 : 8F
DC4B E5 19 DD 23 DD 23 CB 7C : 45
DC53 20 07 B7 ED 42 30 07 E1 : 25
DC5B C9 21 00 00 18 03 60 69 : CE
DC63 2B D1 B7 ED 52 EB C9 DD : 83
DC6B 6E 00 DD 66 01 19 DD 75 : 1D
DC73 00 DD 74 01 DD 23 DD 23 : 52
DC7B C9 21 75 F9 06 05 C5 E5 : 0D
DC83 CD 96 DC E1 C1 30 06 11 : 28
DC8B FC FF 19 10 F1 78 B7 32 : 76
DC93 FD FA C9 4E 23 46 23 EB : 85
DC9B 2A A0 F9 23 23 23 23 B7 : 06
DCA3 ED 42 08 09 08 D8 3E 08 : 66
DCAB 81 4F 3E 00 88 47 2B B7 : BF
-------------------------------------
SUM: 7E 93 21 99 D7 6C 29 25 3FF3

DCB3 ED 42 3F D8 EB 7E 5F 3A : 48
DCBB A2 F9 C6 04 BB D8 57 7B : CA
DCC3 C6 08 BA C9 ED 5B 79 F9 : 0B
DCCB 2A 7D F9 19 22 6D F9 22 : 63
DCD3 71 F9 EB 22 69 F9 22 75 : 70
DCDB F9 ED 5B 7B F9 2A 7F F9 : 57
DCE3 19 22 73 F9 22 77 F9 EB : 24
DCEB 22 6B F9 22 6F F9 C9 CD : A6
DCF3 6C DD 2A 75 F9 ED 5B 69 : 92
DCFB F9 3A 77 F9 47 3A 6B F9 : 88
DD03 4F 3E 08 DD 21 06 FB CD : 61
DD0B CC DE E5 2A 71 F9 ED 5B : 6B
DD13 6D F9 3A 73 F9 47 3A 6F : FC
DD1B F9 4F 3E 08 DD 21 0E FB : 95
DD23 CD CC DE D1 AF DD 21 FE : F3
DD2B FA CD 13 DF 23 E5 2B ED : D9
-------------------------------------
SUM: D1 47 61 16 22 01 CD D5 4ED2

DD33 5B 7F F9 AF DD 21 16 FB : 91
DD3B CD 13 DF 21 79 F9 11 17 : 7A
DD43 FB 01 03 00 ED B0 CD A5 : 0E
DD4B DD C1 0B 78 B1 C8 C5 2A : 89
DD53 02 FB ED 5B 04 FB 19 22 : 7F
DD5B 02 FB DC 89 DD CD 91 DD : 7A
DD63 DD 21 16 FB CD 7A DF 18 : 4D
DD6B DD CD 5A BF 3E FF D3 BD : 90
DD73 3A 53 FA 01 0F 10 67 0F : 1D
DD7B 0F 0F 0F A1 B0 6F 7C A1 : 0A
DD83 B0 67 22 2E FB C9 3A FE : 63
DD8B FA B7 28 08 18 0D 3A FE : 3E
DD93 FA B7 28 07 DD 21 06 FB : DF
DD9B C3 20 DF DD 21 0E FB C3 : 8C
DDA3 20 DF DD E5 E5 D5 C5 2A : 6A
DDAB 0F FB ED 5B 07 FB 3A 11 : 9F
-------------------------------------
SUM: 9D 69 43 E2 9C 27 6C 5A 7261

DDB3 FB 47 3A 09 FB 4F AF DD : 5B
DDBB 21 1E FB CD CC DE E5 ED : 83
DDC3 5B 7D F9 3E 40 DD 21 26 : 73
DDCB FB CD 13 DF 21 17 FB 11 : FE
DDD3 27 FB 01 03 00 ED B0 C1 : 84
DDDB 03 C5 CD FA DD DD 21 1E : 88
DDE3 FB CD 20 DF DD 21 26 FB : E6
DDEB CD 7A DF C1 0B 78 B1 20 : 3B
DDF3 E8 C1 D1 E1 DD E1 C9 D9 : BB
DDFB 2A 29 FB 26 00 ED 5B 27 : E3
DE03 FB 7B CD E8 BE 11 00 40 : 3A
DE0B 19 E6 07 47 CD 98 DE D9 : 69
DE13 C0 E5 D5 C5 2A 21 FB 26 : AB
DE1B 00 ED 5B 1F FB 7B CD E8 : 92
DE23 BE 11 00 40 19 E6 07 4F : 64
DE2B 47 3E 01 28 03 07 10 FD : C5
-------------------------------------
SUM: 4F 22 DF 12 96 84 39 6E FCF8

DE33 5F 79 D9 1E 00 90 30 03 : 92
DE3B ED 44 1C 57 3E 07 D3 BC : 78
DE43 AF D3 BD 3E 80 D3 BC D9 : 65
DE4B 06 02 0E 30 79 C6 02 CD : 54
DE53 28 BF D9 7E 01 BC 04 ED : EC
DE5B 78 CD 7B DE D3 BD 0C 10 : 4A
DE63 F6 D9 3A 5D F9 CD 28 BF : 13
DE6B 7E A3 20 05 79 CD 28 BF : 73
DE73 73 0C 10 D8 C1 D1 E1 C9 : A3
DE7B D9 57 D9 7A B7 D9 7A D9 : 66
DE83 C8 D5 7B B7 D9 7A D9 28 : 23
DE8B 06 0F 15 20 FC D1 C9 07 : E7
DE93 15 20 FC D1 C9 3A 57 FA : 56
DE9B B7 C8 C5 3E 07 D3 BC 78 : 90
DEA3 B7 3E 01 28 03 07 10 FD : 35
DEAB ED 5B 2E FB 4F 3E 33 CD : FE
-------------------------------------
SUM: 9F 62 D7 FC EC 8A 74 ED AA4A

DEB3 28 BF 7A D3 BD 7E DB BC : 06
DEBB A1 47 3E 32 CD 28 BF 7B : 87
DEC3 D3 BD 7E DB BC A1 A0 C1 : A7
DECB C9 DD 73 01 DD 72 02 DD : 48
DED3 71 03 B7 ED 52 30 06 19 : B9
DEDB EB F6 01 ED 52 50 47 7A : 32
DEE3 91 30 04 ED 44 CB C8 5F : E8
DEEB 16 00 78 F6 80 ED 52 08 : 4B
DEF3 19 08 30 03 E6 7F EB EB : 8F
DEFB D5 DD 77 00 23 13 CD 70 : 9C
DF03 BE DD 73 06 DD 72 07 1B : 85
DF0B DD 73 04 DD 72 05 E1 C9 : 52
DF13 B7 ED 52 08 19 08 30 DF : 2E
DF1B F6 80 EB 18 DA DD 4E 00 : 7E
DF23 DD 7E 04 DD 86 06 DD 77 : 1C
DF2B 04 DD 7E 05 DD 8E 07 DD : B3
-------------------------------------
SUM: 7F C6 BA 86 39 73 A5 41 26DD

DF33 77 05 30 08 CD 48 DF CB : 73
DF3B 59 C4 A5 DD C3 42 DF CB : 4E
DF43 79 28 28 18 06 CB 79 28 : 53
DF4B 02 18 20 CB 41 28 0E DD : 59
DF53 6E 01 DD 66 02 2B DD 75 : 31
DF5B 01 DD 74 02 C9 DD 6E 01 : 69
DF63 DD 66 02 23 DD 75 01 DD : 98
DF6B 74 02 C9 CB 49 28 04 DD : 5C
DF73 35 03 C9 DD 34 03 C9 DD : BB
DF7B 4E 00 DD 7E 04 DD 86 06 : 16
DF83 DD 77 04 DD 7E 05 DD 8E : 23
DF8B 07 DD 77 05 08 CB 79 28 : D4
DF93 08 08 38 07 CD 9E DF 18 : B1
DF9B E1 08 D0 CB 71 28 D4 18 : 09
DFA3 BC                      : BC
-------------------------------------
SUM: 17 B6 62 2D C4 98 ED 94 CF4C
```

リスト2 上下・左右反転ルーチン

```
DFC4 CD 05 B0 C8 CD 10 BC CD : B0
DFCC 91 C0 C8 CD 2E A6 38 F7 : E9
DFD4 CD 0A BD 3A 62 F9 E6 01 : 10
DFDC 20 05 CD 2F E0 18 03 CD : E9
DFE4 EE DF CD F2 BC CD C8 A4 : 81
DFEC 18 DD CD 6E DD ED 5B 7B : D0
DFF4 F9 2A 7F F9 45 04 19 C5 : C2
DFFC B5 E5 7B 32 29 FB 7D 32 : 3A
E004 21 FB 2A 79 F9 22 1F FB : F4
E00C 22 27 FB 2A 7D F9 23 E5 : EC
E014 CD FD DD 2A 1F FB 23 22 : 30
E01C 1F FB 22 27 FB E1 2B 7C : E6
E024 B5 20 EC E1 D1 2D 1C C1 : 7D
E02C 10 CD C9 CD 6E DD ED 5B : 06
E034 79 F9 2A 7D F9 44 4D 03 : A6
E03C 19 C5 D5 E5 22 1F FB ED : C1
-----------------------------------
SUM: A5 64 6E 8D 2E E4 77 32 1427

E044 53 27 FB 3A 7B F9 32 21 : 76
E04C FB 32 29 FB 3A 7F F9 47 : 4A
E054 04 C5 CD FD DD 21 21 FB : AD
E05C 34 21 29 FB 34 C1 10 F1 : 6F
E064 E1 D1 2B 13 C1 0B 78 B1 : E5
E06C 20 CF C9                : B8
-----------------------------------
SUM: 87 DF 0E 40 87 65 D4 05 5BA1
```

リスト3 文字ウィンドウ

```
E06F 05 00 01 01 04 01 3F B3 : FE
E077 3B E1 00 01 02 02 03 6E : 92
E07F E1 41 E1 00 01 07 02 08 : 15
E087 8C E1 4F E1 00 03 02 03 : A5
E08F 08 B0 E1 60 E1 00 04 02 : E0
E097 04 08 B5 E1 65 E1 03 77 : 62
E09F 05 05 29 51 02 07 04 04 : 95
E0A7 2B 53 02 70 04 04 2A 52 : 74
E0AF 02 07 07 37 17 48 03 76 : 1F
E0B7 08 38 16 47 03 75 18 48 : 75
E0BF 27 4F 02 70 18 0F 20 52 : 81
E0C7 02 07 07 10 17 20 02 70 : C9
E0CF 04 29 18 2A 01 70 05 0F : F4
E0D7 29 0F 0B 06 00 07 10 08 : 68
E0DF 01 01 09 24 20 20 24 00 : 93
E0E7 08 03 02 09 36 36 1F 1D : BE
-----------------------------------
SUM: 52 E4 46 40 F3 B2 10 AF A55A

E0EF 1D 36 36 00 08 03 04 09 : A1
E0F7 36 36 1F 1D 1D 36 36 00 : 31
E0FF 08 03 06 09 36 36 1F 1D : C2
E107 1D 36 36 00 08 03 08 09 : A5
E10F 36 36 00 08 01 07 09 2E : B3
E117 2C 1F 1D 1D 32 30 00 08 : EF
E11F 03 09 09 28 2B 00 07 00 : 6F

E127 08 02 01 09 EC E7 00 08 : EF
E12F 01 02 09 82 57 1F 1D 1D : 3E
E137 31 36 00 00 11 9D E0 C3 : B8
E13F 3A AE 11 02 01 3A 6C FA : 9C
E147 47 0E 00 3E 01 C3 A5 BC : B8
E14F 11 01 01 3A 6B FA 47 CB : C4
E157 38 E6 01 4F 3E 01 C3 A5 : 15
E15F BC 3A 69 FA 18 03 3A 6A : 18
E167 FA 4F 3E 01 C3 5F BC CD : 33
-----------------------------------
SUM: 97 69 7B C2 9B A6 7F AA AAEE

E16F BB A4 C0 CD 8C A2 11 02 : 2D
E177 01 3A 6C FA 47 AF 4F D5 : BB
E17F CD A5 BC D1 78 32 6C FA : 0F
E187 3E 01 C3 A5 BC CD BB A4 : 8F
E18F C0 CD 8C A2 11 01 01 3A : 08
E197 6B FA 47 CB 38 E6 01 4F : E5
E19F AF D5 CD A5 BC D1 78 87 : 82
E1A7 81 32 6B FA 3E 01 C3 A5 : BF
E1AF BC 21 69 FA 18 03 21 6A : E6
E1B7 FA 3A AC F9 0F D0 E5 CD : 6A
E1BF 8C A2 E1 4E AF E5 CD 5F : 1D
E1C7 BC E1 71 3E 01 C3 5F BC : 2B
-----------------------------------
SUM: 20 30 1D C8 21 84 F6 7C E6E6
```

リスト4 変形部ソースリスト

```
DCF2:               9452:         .LPTON
DCF2:               9453: ;-----------------------------------
DCF2:               9454: ;transform in:(ux,uy),(dux,duy),
DCF2:               9455: ;               (x1,y1),...,(x4,y4)
DCF2:               9456: ;-----------------------------------
DCF2: CD 6C DD      9457: TRANSF: CALL    TRANIC  ;初期設定
DCF5: 2A 75 F9      9458:         LD      HL,(x4)
DCF8: ED 5B 69 F9   9459:         LD      DE,(x1)
DCFC: 3A 77 F9      9460:         LD      A,(y4)
DCFF: 47            9461:         LD      B,A     ;(x4,y4)=(HL,B)
DD00: 3A 6B F9      9462:         LD      A,(y1)
DD03: 4F            9463:         LD      C,A     ;(x1,y1)=(DE,C)
DD04: 3E 08         9464:         LD      A,08H   ;flag初期値=移動と共に描画
DD06: DD 21 06 FB   9465:         LD      IX,TLLINE       ;ポインタ
DD0A: CD CC DE      9466:         CALL    TRSUB   ;左ライン
DD0D: E5            9467:         PUSH    HL      ;Lleft-1=max(lxL-1,lyL-1)
DD0E: 2A 71 F9      9468:         LD      HL,(x3)
DD11: ED 5B 6D F9   9469:         LD      DE,(x2)
DD15: 3A 73 F9      9470:         LD      A,(y3)
DD18: 47            9471:         LD      B,A     ;(x3,y3)=(HL,B)
DD19: 3A 6F F9      9472:         LD      A,(y2)
DD1C: 4F            9473:         LD      C,A     ;(x2,y2)=(DE,C)
DD1D: 3E 08         9474:         LD      A,08H   ;flag=移動と共に描画
DD1F: DD 21 0E FB   9475:         LD      IX,TRLINE
DD23: CD CC DE      9476:         CALL    TRSUB   ;右ライン
DD26: D1            9477:         POP     DE      ;DE=Lleft-1, HL=Lright-1=max(lxR-1,lyR-1)
DD27: AF            9478:         XOR     A       ;flag初期値
DD28: DD 21 FE FA   9479:         LD      IX,TSIDE
DD2C: CD 13 DF      9480:         CALL    TRDSUB  ;左右ライン移動調整設定
DD2F: 23            9481:         INC     HL      ;L++
DD30: E5            9482:         PUSH    HL      ;save maxl=max(Lleft,Lright)
DD31: 2B            9483:         DEC     HL      ;maxl-1;ループカウンタ
DD32: ED 5B 7F F9   9484:         LD      DE,(duy)        ;DE=LY-1
DD36: AF            9485:         XOR     A       ;flag初期値
DD37: DD 21 16 FB   9486:         LD      IX,TDTLN
DD3B: CD 13 DF      9487:         CALL    TRDSUB  ;y方向データ位置移動設定
DD3E: 21 79 F9      9488:         LD      HL,ux
DD41: 11 17 FB      9489:         LD      DE,TDTLN+1
DD44: 01 03 00      9490:         LD      BC,0003H
DD47: ED B0         9491:         LDIR            ;(ux,uy)をワークへ転送
DD49:               9492: TRANS1: ;loop
DD49: CD A5 DD      9493:         CALL    TDRAW   ;1)直線描画
DD4C: C1            9494:         POP     BC      ;pop counter
DD4D: 0B            9495:         DEC     BC      ;counter--
DD4E: 78            9496:         LD      A,B
DD4F: B1            9497:         OR      C
DD50: C8            9498:         RET     Z       ;終了?
DD51: C5            9499:         PUSH    BC
DD52: 2A 02 FB      9500:         LD      HL,(TSIDE+4)    ;Slr
DD55: ED 5B 04 FB   9501:         LD      DE,(TSIDE+6)    ;STlr
DD59: 19            9502:         ADD     HL,DE           ;Slr=Slr+STlr
DD5A: 22 02 FB      9503:         LD      (TSIDE+4),HL    ;Slr
DD5D: DC 89 DD      9504:         CALL    C,TLRSHT        ;左右ラインの短い方
DD60: CD 91 DD      9505:         CALL    TLRLON          ;左右ラインの長い方
DD63: DD 21 16 FB   9506:         LD      IX,TDTLN
DD67: CD 7A DF      9507:         CALL    TRDMOV          ;y方向データ位置移動
DD6A: 18 DD         9508:         JR      TRANS1
DD6C:               9509: ;初期設定(2500用)
DD6C:               9510: ;-------------
DD6C: CD 5A BF      9511: TRANIC: CALL    _GRWMO          ;G-CRTCを単色描画モードに
DD6F: 3E FF         9512:         LD      A,0FFH
DD71: D3 BD         9513:         OUT     (0BDH),A
DD73: 3A 53 FA      9514:         LD      A,(tmcol)       ;透明色
DD76: 01 0F 10      9515:         LD      BC,100FH        ;マスクデータ
DD79: 67            9516:         LD      H,A             ;A=bbbbdddd
DD7A: 0F            9517:         RRCA
DD7B: 0F            9518:         RRCA
DD7C: 0F            9519:         RRCA
DD7D: 0F            9520:         RRCA                    ;A=ddddbbbb
DD7E: A1            9521:         AND     C
DD7F: B0            9522:         OR      B
DD80: 6F            9523:         LD      L,A             ;L=MB30H用カラー
DD81: 7C            9524:         LD      A,H
DD82: A1            9525:         AND     C
DD83: B0            9526:         OR      B
DD84: 67            9527:         LD      H,A             ;H=MB31H用カラー
DD85: 22 2E FB      9528:         LD      (TDRDT+16),HL
DD88: C9            9529:         RET
DD89:               9530: ;左右ラインの移動調整
DD89:               9531: ;-------------------
DD89: 3A FE FA      9532: TLRSHT: LD      A,(TSIDE)       ;短い方
DD8C: B7            9533:         OR      A               ;flag(7bit)=0 なら
DD8D: 28 08         9534:         JR      Z,TLRSLL        ;  左ラインの移動
DD8F: 18 0D         9535:         JR      TLRSLR          ;そうでないなら右ラインの移
動
DD91:               9536:         ;
DD91: 3A FE FA      9537: TLRLON: LD      A,(TSIDE)       ;長い方
DD94: B7            9538:         OR      A               ;flag(7bit)=0 なら
DD95: 28 07         9539:         JR      Z,TLRSLR        ;  右ラインの移動
DD97:               9540:         ;JR      TLRSLL         ;そうでないなら 左ラインの
移動
DD97: DD 21 06 FB   9541: TLRSLL: LD      IX,TLLINE
DD9B: C3 20 DF      9542:         JP      TRMOV           ;描画側の移動
DD9E: DD 21 0E FB   9543: TLRSLR: LD      IX,TRLINE
DDA2: C3 20 DF      9544:         JP      TRMOV           ;描画側の移動
DDA5:               9545: ;直線描画  -----
DDA5:               9546: ;---------------
DDA5: DD E5         9547: TDRAW:  PUSH    IX
DDA7: E5            9548:         PUSH    HL
DDA8: D5            9549:         PUSH    DE
DDA9: C5            9550:         PUSH    BC
DDAA:               9551:         ;initialize
DDAA: 2A 0F FB      9552:         LD      HL,(TRLINE+1)
DDAD: ED 5B 07 FB   9553:         LD      DE,(TLLINE+1)
DDB1: 3A 11 FB      9554:         LD      A,(TRLINE+3)
DDB4: 47            9555:         LD      B,A             ;右ライン(x,y)=(HL,B)
DDB5: 3A 09 FB      9556:         LD      A,(TLLINE+3)
DDB8: 4F            9557:         LD      C,A             ;左ライン(x,y)=(DE,C)
DDB9: AF            9558:         XOR     A               ;flag=移動の際に描画は行わな
い
DDBA: DD 21 1E FB   9559:         LD      IX,TDRDT
DDBE: CD CC DE      9560:         CALL    TRSUB           ;描画側の移動設定
DDC1: E5            9561:         PUSH    HL              ;save l-1=max(lx-1,ly-1)
DDC2:               9562:         ;
DDC2: ED 5B 7D F9   9563:         LD      DE,(dux)        ;LX-1
DDC6: 3E 40         9564:         LD      A,40H           ;flag=x方向のデータ移動
DDC8: DD 21 26 FB   9565:         LD      IX,TDRDT+8
DDCC: CD 13 DF      9566:         CALL    TRDSUB          ;x方向のデータ位置の移動設
定
DDCF: 21 17 FB      9567:         LD      HL,TDTLN+1      ;y方向のデータ位置ワークを
DDD2: 11 27 FB      9568:         LD      DE,TDRDT+9      ;  x方向のデータ位置ワーク
へ
DDD5: 01 03 00      9569:         LD      BC,0003H
DDD8: ED B0         9570:         LDIR                    ;データ転送
DDDA: C1            9571:         POP     BC              ;l-1
DDDB: 03            9572:         INC     BC              ;l=ライン描画用カウンタ
DDDC:               9573: TDRAW1: ;loop
DDDC: C5            9574:         PUSH    BC
DDDD: CD FA DD      9575:         CALL    TRDOT           ;データ位置のカラーを1ドッ
ト描画
DDE0: DD 21 1E FB   9576:         LD      IX,TDRDT
DDE4: CD 20 DF      9577:         CALL    TRMOV           ;描画側の移動
DDE7: DD 21 26 FB   9578:         LD      IX,TDRDT+8
DDEB: CD 7A DF      9579:         CALL    TRDMOV          ;データ側の移動
DDEE: C1            9580:         POP     BC
DDEF: 0B            9581:         DEC     BC
DDF0: 78            9582:         LD      A,B
DDF1: B1            9583:         OR      C
DDF2: 20 E8         9584:         JR      NZ,TDRAW1
DDF4:               9585:         ;
DDF4: C1            9586:         POP     BC
DDF5: D1            9587:         POP     DE
DDF6: E1            9588:         POP     HL
DDF7: DD E1         9589:         POP     IX
DDF9: C9            9590:         RET
DDFA:               9591: ;-----------------------------
DDFA:               9592: ;データ側の位置(X,Y),
DDFA:               9593: ;描画側の位置(x,y)とすると
DDFA:               9594: ; pset(x,y),point(X,Y)を行う   (2500用)
DDFA:               9595: ;-----------------------------
DDFA: D9            9596: TRDOT:  EXX                     ;{exx
DDFB: 2A 29 FB      9597:         LD      HL,(TDRDT+11)
DDFE: 26 00         9598:         LD      H,0
DE00: ED 5B 27 FB   9599:         LD      DE,(TDRDT+9)    ;(X,Y)=(DE,HL)
DE04: 7B            9600:         LD      A,E             ;save (X mod 256)
DE05: CD E8 BE      9601:         CALL    _GADRS          ;Y*40+HL
DE08: 11 00 40      9602:         LD      DE,GRAMP0       ;at 4000H
DE0B: 19            9603:         ADD     HL,DE
DE0C: E6 07         9604:         AND     07H
DE0E: 47            9605:         LD      B,A             ;B=(X mod 8)
DE0F: CD 98 DE      9606:         CALL    TRTMC           ;透明色設定
DE12: D9            9607:         EXX                     ;}
DE13: C0            9608:         RET     NZ              ;透明色だったら
DE14:               9609:         ;                       ;描画せずに抜ける
DE14: E5            9610:         PUSH    HL
DE15: D5            9611:         PUSH    DE
DE16: C5            9612:         PUSH    BC
DE17: 2A 21 FB      9613:         LD      HL,(TDRDT+3)
DE1A: 26 00         9614:         LD      H,0
DE1C: ED 5B 1F FB   9615:         LD      DE,(TDRDT+1)    ;(x,y)=(DE,HL)
DE20: 7B            9616:         LD      A,E
DE21: CD E8 BE      9617:         CALL    _GADRS          ;y*40+x
```

```
DE24: 11 00 40          9618:         LD      DE,GRAMP0          ;at 4000H
DE27: 19                9619:         ADD     HL,DE
DE28: E6 07             9620:         AND     07H
DE2A: 4F                9621:         LD      C,A                ;C=(x mod 8)
DE2B: 47                9622:         LD      B,A                ;B=(x mod 8)
DE2C: 3E 01             9623:         LD      A,01H    ;dot
DE2E: 28 03             9624:         JR      Z,TRDOT2
DE30: 07                9625: TRDOT1: RLCA
DE31: 10 FD             9626:         DJNZ    TRDOT1
DE33: 5F                9627: TRDOT2: LD      E,A                ;E=ドット用ビットデータ
DE34: 79                9628:         LD      A,C                ;x mod 8
DE35: D9                9629:         EXX              ;{exx
DE36: 1E 00             9630:         LD      E,0                ;flag=ビットシフト(左右どち
らか)
DE38: 90                9631:         SUB     B                  ;(x mod 8)-(X mod 8)
DE39: 30 03             9632:         JR      NC,TRDOT3
DE3B: ED 44             9633:         NEG                        ;abs(..)
DE3D: 1C                9634:         INC     E                  ;<- へ shift
DE3E: 57                9635: TRDOT3: LD      D,A                ;D'=bit_shift_counter
DE3F:                   9636:         ;
DE3F: 3E 07             9637:         LD      A,07H              ;
DE41: D3 BC             9638:         OUT     (0BCH),A           ;BRGI4プレーン
DE43: AF                9639:         XOR     A                  ;同時読み込みモードに
DE44: D3 BD             9640:         OUT     (0BDH),A           ;設定
DE46: 3E 80             9641:         LD      A,80H              ;同時書き込みモードに
DE48: D3 BC             9642:         OUT     (0BCH),A           ;設定
DE4A: D9                9643:         EXX              ;}
DE4B:                   9644:         ;loop
DE4B: 06 02             9645:         LD      B,2
DE4D: 0E 30             9646:         LD      C,MPGP0            ;MB30H
DE4F: 79                9647: TRDOT5: LD      A,C
DE50: C6 02             9648:         ADD     A,2                ;MB32H(裏画面)
DE52: CD 28 BF          9649:         CALL    _SET4H             ;4000Hに配置
DE55: D9                9650:         EXX              ;{exx
DE56: 7E                9651:         LD      A,(HL)             ;データ位置(X,Y)アドレス
DE57: 01 BC 04          9652:         LD      BC,04BCH           ;B,R,G,I毎に
DE5A: ED 78             9653: TRDOT6: IN      A,(C)              ;データを得る point(X,Y)
DE5C: CD 7B DE          9654:         CALL    TRBST              ;得られたデータを描画位置へ
シフト
DE5F: D3 BD             9655:         OUT     (0BDH),A           ;データをG-CRTCへ設定
DE61: 0C                9656:         INC     C
DE62: 10 F6             9657:         DJNZ    TRDOT6
DE64: D9                9658:         EXX              ;}
DE65: 3A 5D F9          9659:         LD      A,(MP4)
DE68: CD 28 BF          9660:         CALL    _SET4H             ;マスキングVRAM
DE6B: 7E                9661:         LD      A,(HL)             ;を読み込む
DE6C: A3                9662:         AND     E                  ;ドット用ビットデータとAND
DE6D: 20 05             9663:         JR      NZ,TRDOT7          ;マスキングされていたら
DE6F:                   9664:         ;draw
DE6F: 79                9665:         LD      A,C
DE70: CD 28 BF          9666:         CALL    _SET4H
DE73: 73                9667:         LD      (HL),E             ;GRAMへ書き込む
DE74: 0C                9668: TRDOT7: INC     C
DE75: 10 D8             9669:         DJNZ    TRDOT5
DE77:                   9670:         ;
DE77: C1                9671:         POP     BC
DE78: D1                9672:         POP     DE
DE79: E1                9673:         POP     HL
DE7A: C9                9674:         RET
DE7B:                   9675: ;bit shift------
DE7B:                   9676: ;---------------
DE7B: D9                9677: TRBST:  EXX              ;}
DE7C: 57                9678:         LD      D,A      ;data save
DE7D: D9                9679:         EXX              ;{exx
DE7E: 7A                9680:         LD      A,D                ;シフトカウンタ
DE7F: B7                9681:         OR      A
DE80: D9                9682:         EXX              ;}
DE81: 7A                9683:         LD      A,D
DE82: D9                9684:         EXX              ;{exx
DE83: C8                9685:         RET     Z
DE84:                   9686:         ;
DE84: D5                9687:         PUSH    DE
DE85: 7B                9688:         LD      A,E                ;左か右かのflag
DE86: B7                9689:         OR      A
DE87: D9                9690:         EXX              ;}
DE88: 7A                9691:         LD      A,D
DE89: D9                9692:         EXX              ;{exx
DE8A: 28 06             9693:         JR      Z,TRBSTL
DE8C:                   9694: TRBSTR: ;<-shift
DE8C: 0F                9695:         RRCA
DE8D: 15                9696:         DEC     D
DE8E: 20 FC             9697:         JR      NZ,TRBSTR
DE90: D1                9698:         POP     DE
DE91: C9                9699:         RET
DE92:                   9700: TRBSTL: ;->shift
DE92: 07                9701:         RLCA
DE93: 15                9702:         DEC     D
DE94: 20 FC             9703:         JR      NZ,TRBSTL
DE96: D1                9704:         POP     DE
DE97: C9                9705:         RET
DE98:                   9706: ;透明色チェック--
DE98:                   9707: ;---------------
DE98: 3A 57 FA          9708: TRTMC:  LD      A,(tmflag)         ;透明色モードで
DE9B: B7                9709:         OR      A                  ;ないなら
DE9C: C8                9710:         RET     Z                  ;抜ける
DE9D:                   9711:         ;
DE9D: C5                9712:         PUSH    BC
DE9E: 3E 07             9713:         LD      A,7
DEA0: D3 BC             9714:         OUT     (0BCH),A
DEA2: 78                9715:         LD      A,B                ;A=(X mod 8)
DEA3: B7                9716:         OR      A
DEA4: 3E 01             9717:         LD      A,01H
DEA6: 28 03             9718:         JR      Z,TRTMC2
DEA8: 07                9719: TRTMC1: RLCA                       ;データ位置のビットデータ
DEA9: 10 FD             9720:         DJNZ    TRTMC1
DEAB: ED 5B 2E FB       9721: TRTMC2: LD      DE,(TDRDT+16)      ;透明色
DEAF: 4F                9722:         LD      C,A                ;データ位置のビットデータ
DEB0: 3E 33             9723:         LD      A,MPGP1+1          ;MB33H(裏画面)
DEB2: CD 28 BF          9724:         CALL    _SET4H
DEB5: 7A                9725:         LD      A,D                ;カラー設定
DEB6: D3 BD             9726:         OUT     (0BDH),A
DEB8: 7E                9727:         LD      A,(HL)
DEB9: DB BC             9728:         IN      A,(0BCH)           ;データ読み込む
DEBB: A1                9729:         AND     C                  ;データ位置のビットデータで
AND
DEBC: 47                9730:         LD      B,A
DEBD:                   9731:         ;
DEBD: 3E 32             9732:         LD      A,MPGP1            ;MB32H(裏画面)
DEBF: CD 28 BF          9733:         CALL    _SET4H
DEC2: 7B                9734:         LD      A,E                ;カラー設定
DEC3: D3 BD             9735:         OUT     (0BDH),A
DEC5: 7E                9736:         LD      A,(HL)
DEC6: DB BC             9737:         IN      A,(0BCH)
DEC8: A1                9738:         AND     C                  ;データ位置のビットデータで
AND
DEC9: A0                9739:         AND     B                  ;両方のカラーでANDして256色
にしている
DECA: C1                9740:         POP     BC
DECB: C9                9741:         RET
DECC:                   9742: ;---------------------------
DECC:                   9743: ;描画位置(DE,C)から(HL,B)へ
DECC:                   9744: ;移動する際の設定
DECC:                   9745: ; in: IX(ポインタ),(DE,C),(HL,B),A(flag)
DECC:                   9746: ;out: HL(=max(lx-1,ly-1))
DECC:                   9747: ;---------------------------
DECC: DD 73 01          9748: TRSUB:  LD      (IX+1),E           ;DE
DECF: DD 72 02          9749:         LD      (IX+2),D           ;最初の位置をワークへ
DED2: DD 71 03          9750:         LD      (IX+3),C           ;B
DED5: B7                9751:         OR      A
DED6: ED 52             9752:         SBC     HL,DE              ;HL-DE
DED8: 30 06             9753:         JR      NC,TRSUB1
DEDA: 19                9754:         ADD     HL,DE              ;HL<DE
DEDB: EB                9755:         EX      DE,HL              ;HL>DEにして
DEDC: F6 01             9756:         OR      01H                ;flag dx=-1
DEDE: ED 52             9757:         SBC     HL,DE              ;HL=abs(HL-DE)
DEE0:                   9758: TRSUB1: ;                          ;HL=lx-1
DEE0: 50                9759:         LD      D,B                ;D=B
DEE1: 47                9760:         LD      B,A                ;B=flags
DEE2: 7A                9761:         LD      A,D
DEE3: 91                9762:         SUB     C                  ;D-C
DEE4: 30 04             9763:         JR      NC,TRSUB2
DEE6: ED 44             9764:         NEG                        ;A=abs(D-C)
DEE8: CB C8             9765:         SET     1,B                ;flag dy:=-1
DEEA: 5F                9766: TRSUB2: LD      E,A
DEEB: 16 00             9767:         LD      D,0                ;DE=ly-1
DEED: 78                9768:         LD      A,B                ;A= flags
DEEE: F6 80             9769:         OR      80H                ;flag XLINE(lx>=ly)
DEF0: ED 52             9770:         SBC     HL,DE              ;(lx-1)-(ly-1)
DEF2: 08                9771:         EX      AF,AF'
DEF3: 19                9772:         ADD     HL,DE
DEF4: 08                9773:         EX      AF,AF'
DEF5: 30 03             9774:         JR      NC,TRSUB3
DEF7: E6 7F             9775:         AND     7FH                ;flag YLINE(lx<ly)
DEF9: EB                9776:         EX      DE,HL              ;DE>HLに
DEFA: EB                9777: TRSUB3: EX      DE,HL              ;DE>HL
DEFB: D5                9778:         PUSH    DE                 ;max(lx-1,ly-1)
DEFC: DD 77 00          9779:         LD      (IX+0),A
DEFF: 23                9780:         INC     HL                 ;lx,lyにする
DF00: 13                9781:         INC     DE
DF01: CD 70 BE          9782:         CALL    _XDIV              ;DE=10000H*HL/DE
DF04: DD 73 06          9783:         LD      (IX+6),E           ;ST
DF07: DD 72 07          9784:         LD      (IX+7),D
DF0A: 1B                9785:         DEC     DE
DF0B: DD 73 04          9786:         LD      (IX+4),E           ;S=ST-1
DF0E: DD 72 05          9787:         LD      (IX+5),D
DF11: E1                9788:         POP     HL                 ;max(lx-1,ly-1)
DF12: C9                9789:         RET
DF13:                   9790: ;---------------------------
DF13:                   9791: ;データの長さHLとDEでの
DF13:                   9792: ;移動する際の設定
DF13:                   9793: ; in: IX(ポインタ),(HL,DE),A(flag)
DF13:                   9794: ;out: HL(=max(lx-1,ly-1))
DF13:                   9795: ;---------------------------
DF13: B7                9796: TRDSUB: OR      A
DF14: ED 52             9797:         SBC     HL,DE
DF16: 08                9798:         EX      AF,AF'
DF17: 19                9799:         ADD     HL,DE
DF18: 08                9800:         EX      AF,AF'
DF19: 30 DF             9801:         JR      NC,TRSUB3
DF1B: F6 80             9802:         OR      80H                ;flag HL<DE
DF1D: EB                9803:         EX      DE,HL
DF1E: 18 DA             9804:         JR      TRSUB3
DF20:                   9805: ;描画側の移動
DF20:                   9806: ;---------------
DF20: DD 4E 00          9807: TRMOV:  LD      C,(IX+0)           ;flags
DF23: DD 7E 04          9808:         LD      A,(IX+4)
DF26: DD 86 06          9809:         ADD     A,(IX+6)
DF29: DD 77 04          9810:         LD      (IX+4),A
DF2C: DD 7E 05          9811:         LD      A,(IX+5)
DF2F: DD 8E 07          9812:         ADC     A,(IX+7)
DF32: DD 77 05          9813:         LD      (IX+5),A           ;S=S+ST
DF35: 30 08             9814:         JR      NC,TRMOV1
DF37: CD 48 DF          9815:         CALL    TRMVS              ;短い方の移動
DF3A: CB 59             9816:         BIT     3,C                ;flag 移動と同時に直線描画?
DF3C: C4 A5 DD          9817:         CALL    NZ,TDRAW           ;直線描画
DF3F: C3 42 DF          9818: TRMOV1: JP      TRMVL              ;長い方の移動
DF42:                   9819: ;
DF42:                   9820: TRMVL:  ;long
DF42: CB 79             9821:         BIT     7,C
DF44: 28 28             9822:         JR      Z,TRMVY            ;y 移動
DF46: 18 06             9823:         JR      TRMVX              ;x 移動
DF48:                   9824: TRMVS:  ;short
DF48: CB 79             9825:         BIT     7,C
DF4A: 28 02             9826:         JR      Z,TRMVX            ;x 移動
DF4C: 18 20             9827:         JR      TRMVY              ;y 移動
DF4E:                   9828: ;
DF4E: CB 41             9829: TRMVX:  BIT     0,C
DF50: 28 0E             9830:         JR      Z,TRMVX1
DF52:                   9831:         ;x--                       ;x --
DF52: DD 6E 01          9832:         LD      L,(IX+1)
DF55: DD 66 02          9833:         LD      H,(IX+2)
DF58: 2B                9834:         DEC     HL
DF59: DD 75 01          9835:         LD      (IX+1),L
DF5C: DD 74 02          9836:         LD      (IX+2),H
DF5F: C9                9837:         RET
DF60:                   9838: TRMVX1: ;x++                       ;x ++
DF60: DD 6E 01          9839:         LD      L,(IX+1)
DF63: DD 66 02          9840:         LD      H,(IX+2)
DF66: 23                9841:         INC     HL
DF67: DD 75 01          9842:         LD      (IX+1),L
DF6A: DD 74 02          9843:         LD      (IX+2),H
DF6D: C9                9844:         RET
DF6E:                   9845: ;
DF6E: CB 49             9846: TRMVY:  BIT     1,C
DF70: 28 04             9847:         JR      Z,TRMVY1
DF72:                   9848:         ;y--                       ;y --
DF72: DD 35 03          9849:         DEC     (IX+3)
DF75: C9                9850:         RET
DF76:                   9851: TRMVY1: ;y++                       ;y ++
DF76: DD 34 03          9852:         INC     (IX+3)
DF79: C9                9853:         RET
DF7A:                   9854: ;データ側の移動
DF7A:                   9855: ;---------------
DF7A: DD 4E 00          9856: TRDMOV: LD      C,(IX+0)           ;flags
DF7D: DD 7E 04          9857: TRDMV1: LD      A,(IX+4)
DF80: DD 86 06          9858:         ADD     A,(IX+6)
DF83: DD 77 04          9859:         LD      (IX+4),A
DF86: DD 7E 05          9860:         LD      A,(IX+5)
DF89: DD 8E 07          9861:         ADC     A,(IX+7)
DF8C: DD 77 05          9862:         LD      (IX+5),A           ;Sd=Sd+STd
DF8F: 08                9863:         EX      AF,AF'  ;{af       ;save CYフラグ
DF90: CB 79             9864:         BIT     7,C
DF92: 28 08             9865:         JR      Z,TRDMV3
DF94:                   9866:         ;l<L                       ;描画側<データ側
DF94: 08                9867:         EX      AF,AF'  ;af}
DF95: 38 07             9868:         JR      C,TRDMV2           ;CYがたつまで
DF97: CD 9E DF          9869:         CALL    TRDMSB             ;データ位置の
DF9A: 18 E1             9870:         JR      TRDMV1             ;移動を繰り返す
DF9C:                   9871: ;TRDMV2: CALL    TRDMSB
DF9C:                   9872: ;        RET
DF9C:                   9873: TRDMV3: ;l>=L                      ;描画側≧データ側
DF9C: 08                9874:         EX      AF,AF'  ;af}
DF9D: D0                9875:         RET     NC
DF9E:                   9876:        ;CALL    TRDMSB
DF9E:                   9877:        ;RET
DF9E:                   9878: TRDMV2:                            ;データ位置の移動
DF9E: CB 71             9879: TRDMSB: BIT     6,C
DFA0: 28 D4             9880:         JR      Z,TRMVY1           ;x 方向++
DFA2:                   9881:         ;xmov
DFA2: 18 BC             9882:         JR      TRMVX1             ;y 方向++
```

SOFTWARE INFORMATION

SOFTWARE INFORMATION

X1/X1turbo
- 維新の嵐
- 麻雀狂時代SPECIALⅡ・冒険編
- 第4のユニット3・デュアルターゲット

X68000
- ジャック・ニクラウス・チャンピオンシップ・ゴルフ
- スターシッププランデブー
- ミッド・ガルツ68Kゴールド
- ダブルイーグル
- 38万キロの虚空
- 高速日本語マルチスクリーンエディタJAM
- ES68K

おー，バイクがコーナーをガンガン攻めながら走る走る。とにかくこのスピード感は凄い。グランプリレーサーの爽快感を味わうにうってつけのゲームといえそう。移植はお馴染みのSPS。発売が待ち遠しいソフトですね。下の写真はブローダーバンド・ジャパンが現在進めてくれているウィングス。ゼロ戦で急降下爆撃や空中戦が楽しめる，ちょっと毛色の変わったシューティングゲームなのです。

話題のソフトウェア

スーパーハングオン登場

さて，この秋に登場するX68000用ゲームソフトのなかで，話題を集めそうな最新ソフトはこれだ！　というわけで，まずは写真を見てください。なんと，いきなりシャープから**スーパーハングオン**が登場だぁ。

とにかく，風を切ってコーナーに飛び込むスピード感を完璧に再現。もう気分はすっかりサロンかレイニーか。詳細は来月，サンダーブレードとともにお届けするから，楽しみに待っててね。そうそう，それと電波新聞社のファンタジーゾーンがようやく発売になりました。こちらはGAME REVIEWで紹介しているから，そっちを見ね。

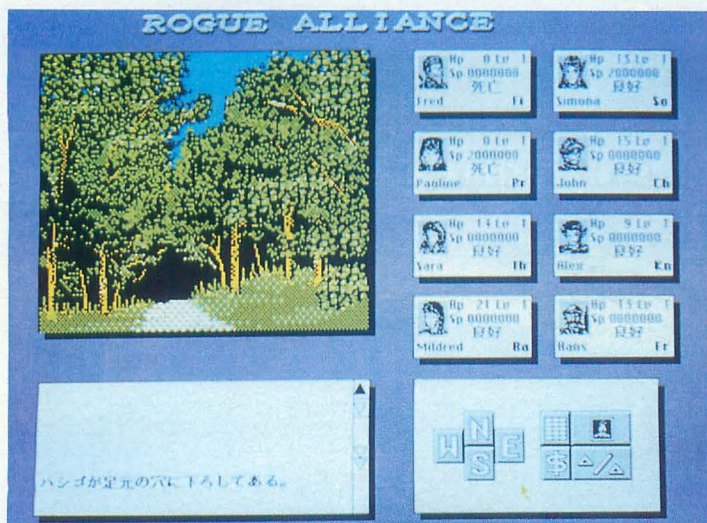
その下で巨大空母を相手に戦うゼロ戦の写真は，ブローダーバンド・ジャパンが現在進めてくれている**ウィングス**の雄姿。こ

X68000のアクションゲーム強し！

		(前回順位)
1.	アフターバーナー	1
2.	ジェノサイド	—
3.	TETRIS (X1/X68000)	6
4.	R-TYPE	10
5.	アドヴァンスト・ファンタジアン	2
6.	ソーサリアン (追加シナリオ含む)	15
7.	Might and MagicⅡ	9
8.	デス・ブリンガー	12
9.	野球道	5
10.	信長の野望・戦国群雄伝	4

今月から，この「読者が選ぶ今月のゲーム10」担当が代わったので，ちょっと私なりにスタイルを変えてみました。これからもよろしくお願いします。というわけで，やっぱりアフターバーナーは強いですね。でも，見てくださいこのジェノサイド。ハガキが届いた時点ではまだ発売されていなかったのに，堂々と2位に入ってしまいました。これで，X68000のラインアップはますます強力になったような気がします。

あとは追加シナリオの発売が決定したソーサリアンが，また帰ってきました。これはもうX1 turboユーザーの定番商品といえそうですね。

さて，来月の見所といえば
・ジェノサイドは果たしてトップを狙えるか？
・ファンタジーゾーンの動向は？
といったところでしょうか。皆さんからのおハガキをお待ちしています。　(浦)

ここに並んだローグ・アライアンス，サバッシュ，ねじ式などは，来月のゲームの特集に登場の予定。アクションあり，RPG, AVGとまたまたなんでもありの特集となってしまいそう。右上の写真は，今回，中森氏がmFORTHを使って作ってくれたサンプル写真なのです。

れまでのものとは違って，フライトシミュレータ＋シューティングアクションという，一風変わったゲームとなりそうなので，期待できそう。

それでは，この8月から秋にかけてX68000に登場予定の新作をここでズラッとご紹介していきたいと思いますので，ゲームファンの皆さんは心して読むよーに（X1ユーザーの皆さんごめんなさい）。まずは，日本テレネットの**夢幻戦士ヴァリスII**。ウルフ・チームは，この8月に一挙に2本ミッド・**ガルツゴールド68K**と**バルセロナの風**。ホビージャパンの**リングマスター**，バショウハウスの**エメラルド・ドラゴン**，スタジオパンサーの**JOSHUA**，日本コンピュータシステムの**テンプルマスター**，ザイン・ソフトは**雪の国クルージュ**，**バルーサの復讐**，ハミングバードソフトの**ロードス島戦記**，アートディンクの**ダブルイーグル**にリバーヒルソフトの**D.C.コネクション**。それからシャープの**サンダーブレード**に**スーパーハングオン**，コナミの**A-JAX**もあるぞっ！どうだ，一気にこれだけ並べると，見事にこれまでの情報と重複してしまったから，どれが初めて聞く最新作かどうかすっかりわからなくなってしまっただろう。それは紹介している側だってよく覚えていないのだから，仕方がない。ましてや発売日が正式に決まっていないものが山ほどあるんだから，誰も整理できるわけがない。

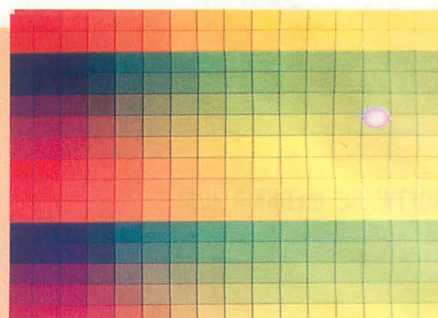
しかしもっと凄いのは，ゲームタイトルこそ発表されていないが，X68000に年内発売予定のソフトハウスが，まだまだこのほかにもごろごろしているのだ。どうだ，まいったか！

というわけで，来月はゲーム特集をやるから，期待して待っているように！

サイクロンで作ったCG・アニメ大募集

サイクロン68K/Express68および，アニメキットを使っているユーザーを対象に，発売元のアンス・コンサルタンツが10月7日，東京渋谷の新大宗ビルフォーラム8において，**「第1回サイクロンCG大会」**を開催します。

参加資格は，X68000のサイクロンユーザーであれば，それでOK。応募方法は，当日会場受け付けに午後2時までに提出するか，または9月30日（当日消印有効）までにアンス・コンサルタンツ本社までデータの入ったフロッピーディスクを郵送してください。応募者全員には，大会記念オリジナルTシャツがプレゼントされるそうです。

このコンテストで選ばれた優秀者には，グランプリ賞1名に賞金20万円と賞状が，アニメ，モデリング，レンダリングの各部門賞それぞれ1名には賞金10万円と賞状が贈られます。このほか，なんと「Oh!X賞」，「LOGiN賞」，「シャープ賞」各2名にも豪華賞品が用意されるんだそうです（Oh!Xの賞品はまだ決めていないけど）。こんな機会って，そうあるものじゃありませんから皆さんふるってご応募くださいね。

〈問い合わせ先〉　アンス・コンサルタンツ　☎092(522)6347

X1/X1turbo用新作ソフト

☆……8月1日現在発売中　★……近日発売予定
＊明記されたもの以外の価格については消費税は含まれておりません

★維新の嵐

1858年。浦賀にペリー総督率いる4隻の艦隊が来航した。ちょうどそのころ，経済情勢が不安定な日本は，これを発端に一気に激動の時代を迎えることになる。そしてそこに，日本の将来を変えようと志に燃える英雄たちがいた。

彼らの持つ思想と，そしてそれらにまつわる人物たちをテーマに展開される，戦国シミュレーションシリーズ「維新の嵐」。500人にものぼる英雄たちは，日本の政治をどうしてゆくかという思想のもと立ち上がろうとしている。プレイヤーも歴史上の人物となって，周囲を説得し，藩主を説き，ときには剣の力でねじ伏せて，国内思想の統一を目指す。君は，この激流のなかでどう生きるか？

なお，この維新の嵐は，光栄のオリジナルCD付きのサウンドウェアバージョンも同時に発売される。

X1turbo用　5"2D版3枚組　9,800円
CD付きサウンドウェア　12,200円
（2ドライブ専用）
光栄　☎044(61)6861

★麻雀狂時代SPECIAL II・冒険編

麻雀狂時代スペシャルに，パソコン版オリジナルの続編が登場。2人対戦モードは前作のような単なる麻雀大会ではなく，今度はストーリーまでついての登場だ。その物語とは，彼女の「今日子」を取り戻すため，情報をもとめてさすらう主人公。持ち金に応じたレートでかけ麻雀をし，情報を得ていく。積み込みはもちろん，雀荘で購入したアイテムによって相手の少牌を誘ったりもできる。

4人対局では，プレイヤーの振込み回数，トップ回数，上がり役の記録，解析が可能という豪華セット。前作同様7人の美女も登場するが，それ以外の新しい趣向でも十分に楽しませてくれる麻雀ゲームとなっている。

X1turbo用　5"2D版3枚組　6,800円
（2ドライブ専用）

マイクロネット　☎011(561)1370

☆第4のユニット3・デュアルターゲット

ブロンウィンのキャラクターと，アニメ番組形式の手軽なストーリー展開が人気の第4のユニットシリーズ。その最新作，「デュアルターゲット」がいよいよX1turboに登場だ。

今回のストーリーは，越中博士とサイコパワーブースターが奪われ，ブロンウィン，アッシュたちはその奪回の任務に就く。そこでブロンウィンは自分とまったく同じ姿をした敵に遭遇する。もしや彼女は，自分のオリジナルでは？　戦いつつも苦悩するブロンウィン。果たしてその謎が解き明かされるとき，彼女がそこに見た真実とはいったい……。

前2作とは変わって，感情的要素をコマンドとして与える新しい手法を用いており，より感情移入を強く打ち出した構成となっている。このストーリーを体験し終えたとき，君はブロンウィンの秘密を知ることになる。

X1turbo用　5″2D版3枚組　7,800円
（2ドライブ専用）
データウエスト　☎06(968)1236

X68000ソフト&ツールズ

★ジャック・ニクラウス・チャンピオンシップ・ゴルフ

ジャック・ニクラウスと名を聞けば，アメリカ4大メジャーを制覇した偉大なプレイヤーの側面が頭に浮かぶ。だが彼がゴルフコースのデザイナーとしても優秀であることは，一般の人にはあまり馴染みのない話だろう。

その彼がデザイナーとしてのセンスを生かし，メジャー大会の舞台の中から厳選したのが，この「ジャック・ニクラウス・チャンピオンシップ・ゴルフ」だ。

プレイヤーはストローク，もしくはスキンズプレイでホールをまわる。人間同士のほかにコンピュータに登録されたゴルファーとのプレイも可能。そのなかにはジャック・ニクラウス本人も入っている。このゲームは，ショットの練習だけでなく各ホールを練習ラウンドとしてまわることもできる。画面は3Dアングルによるもので，ショットのアニメーションはデジタイズしたデータを基に作成したという，これまでにない凝った手法をとっている。

日本版の製作はビクター音楽産業。X68000のゴルフゲームのラッシュのなかで，海外ソフトからの移植の持ち味をしっかりと出してくれることを期待したい。

X68000用　5″2HD版　価格未定
ビクター音楽産業　☎03(423)7901

☆スターシップランデブー

宇宙空間に突如出現した，謎の“穴”と正体不明の宇宙船。貴方はこの空間の謎を解くため，単独潜入を試みた。その船内で見たものは……。

ステージ前半はアクションゲーム。敵ロボットをかわしながら船内に散らばっているアイテムを集め，乗員を探し出すのだ。そして，見つけ出した女戦士を尋問するのが，お楽しみスペシャルステージ。マウスで体をこちょこちょして情報を聞き出す。4人の女兵士を制覇したとき，この宇宙船の正体が明らかになるのだが。制覇するのがメインかな，やっぱり。

特にこのX68000版は，デジタイズ画面とPCMサウンドがゲームをより一層盛り上げてくれている（？）。まっ，詳しいことは自分の目で確かめるのが一番なんでしょうね。きっと。

X68000用　5″2HD版2枚組　8,800円
スキャップトラスト　☎03(407)0687

★ミッド・ガルツ68Kゴールド

ホワイトドラゴンに育てられた少年カイン・サージュと，その友だちの竜サークン・ホワイト。2人がある日洞窟のなかで妖しい1本の剣を見つけたところから，この天界をも揺るがす壮大なドラマはスタートする。

ウルフ・チームの意欲作「ミッド・ガルツ」は，全12話から成るアクションアドベンチャーだ。横スクロール型のシューティングが主体だが，人物やイベントに出会うとビジュアルシーンが展開される。ビジュアルにこだわるウルフ・チームの作品だけに，X68000でどんな世界が展開されるか楽しみな作品だ。

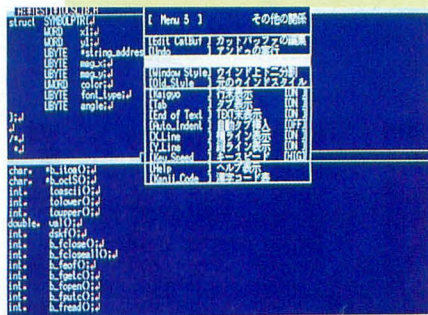
X68000用　5″2HD版4枚組　12,800円
ウルフ・チーム　☎03(5273)4795

★ダブルイーグル

「えっ？　アートディンクからゴルフゲーム？」そう，A列車や大海令など，独特のセンスにファンの多いアートディンクの最新作が，この「ダブルイーグル」だ。

今回もそのオリジナリティがいかんなく発揮され，タイミングさえ同じなら，ボールがころがるところも一緒という，パターン化された従来のゴルフゲームとは一線を画し，体力，気力，技術，疲労の4つのパラメータの組み合わせが，プレイに大きく影響してくる。ゲームの進行は，まったくの素人がプロテストを受け出場資格を得，やがては40もあるトーナメントを選んで出場し優勝を狙うという，太陽君も真っ青のゴルフ人生を歩んでいくのだ。引退セレモニーでは，プレイヤーが歩んだゴルファー人生を振り返るダイジェスト機能も付いている。

X68000用　5″2HD版2枚組　9,500円
アートディンク　☎0474(77)7541

ミッド・ガルツゴールド68K

JAMES68K

★38万キロの虚空

ノベルウェアシリーズ第4弾が，X68000から登場する。今度は「ソフトで〜」とはうって変わって，ドームの流れをくむSFサスペンス。

物語は，21世紀半ば人類はいよいよ宇宙植民時代を迎えようとしていた。米国元大統領クロフォード率いるNSCA財団が計画を進めたスペースコロニーが完成を迎えることになった。主人公相場謙は，コロニーの取材班として現地へ出向くのだが，オープニングセレモニーを前に漂う陰謀の影。目の前に現れてくる事態に対して，どうふるまうかはすべてプレイヤー次第。主人公の立場に合わせたストーリーがそれぞれ用意されているのだ。

雇用法，亜欧の経済格差，環境問題，コンピュータ通信網，人口問題，国際政治情勢など，複雑に絡み合うさまざまな要素のなかでドラマは展開していく。

X68000用　5″2HD版4枚組　9,800円
システムサコム　☎03(635)5145

☆高速日本語マルチスクリーンエディタJAMES68K

エディタは反応速度が命という人にとって，決定版ともいうべき高速マルチスクリーンエディタが登場した。

ウィンドウ数は50，マークジャンプにより異なるウィンドウ間の参照が一発で行えるほか，ディスクの階層ディレクトリやファイル内容の参照により，ファイルの選択も手間をかけずにできる。仕様としてはほぼ標準というところだが，なによりもJAMES68Kの特長は，ほかに例を見ない操作性と高速性だろう。

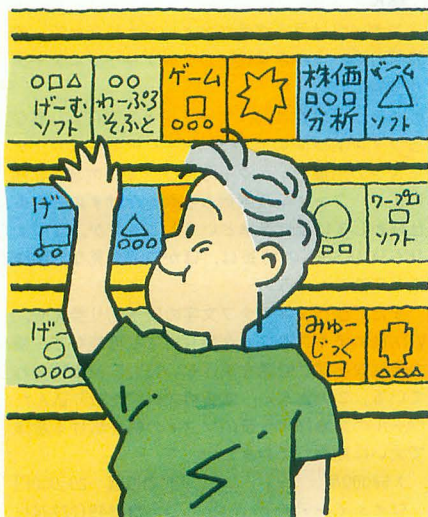
また，改行文字やタブ文字の表示切り換えや，VRAMのテキストワーク割り当てが可能など，極力無駄な手間と時間をかけない設計に仕上げられている。このように，低価格で機能を絞り込んだツールの登場は，今後のソフトの幅を広げる意味で大いに期待しておきたい。

X68000用　5″2HD版　20,000円
ワイ・イー・ティ　☎0849(22)2411

第4のユニット3・デュアルターゲット

スターシップランデブー

GAME REVIEW

今月の目玉はやはりX68000版のファンタジーゾーンということになるんでしょうね。それと，X1版麻雀狂時代SPECIALⅡと，ちょっぴりエッチなX68000版スターシップランデブーという，かなり異色な組み合わせでお届けします。

麻雀狂時代SPECIALⅡ・冒険編

連れ去られた恋人を探してさすらう雀士。暗い過去を引きずりながら，彼は今日も雀荘で戦いを繰り広げるのです。

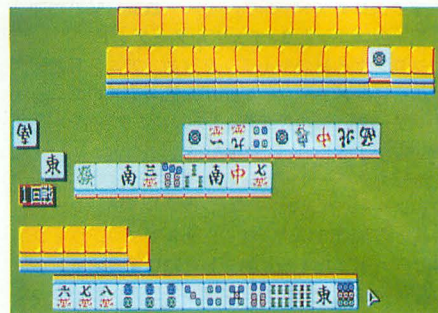
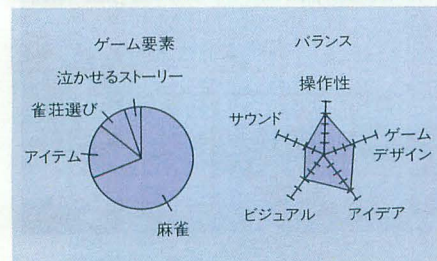
▶おぉーっ，コスプレの娘がミサイルでパーツを吹き飛ばして脱ぐぞーっ！　この演出，洗練されていないグラフィック，音声合成，今回もマイクロネットのセンスが大爆発だぁ！

みんな変に“良い子”ソフトを目指すなかで，わざとクセのある強力ソフトを作るこの姿勢がいい。その証拠にアイテム購入や役の記録，ストーリーなど最近のトレンドはすべて押さえてある。しかもストーリーが奪われた恋人を探すときたもんだ。このたまらないB級ウケ狙いが，非常に素晴らしい。

マウスでの操作性はいいんだけど，このキーボードのほうは，決定/取り消しがなんだか逆なんじゃないか，と思えるのが残念。

万人ウケはしないだろうけど，僕は好きだなぁ。同じ北海道のこぎれいな某まあじゃんゲームを慶応とするなら，こっちは早稲田って感じだよね。

熱中度▶▶▶▶▶▷▷　(H.U.)

▶このゲームが前作と大きく違うところは，やっぱりゲームに出てくる女の子がそれらしくなったことでしょうね。2人麻雀のときに，まるでどこかのアドベンチャーゲームみたいにマップが出てきてどの店で麻雀をするか決めるんですが，その決めた場所で対戦相手の女の子が「よろしくー」と挨拶したり，タンカをきったりしてくれたりという演出があるわけです。

ゲームセンターの麻雀ゲームが最近，女の子とのストーリーが付いたものが出てきた（掟とかP3とかね）けど，これはその影響なんでしょうか。コンピュータの思考ルーチンもこれ以上強くすると問題がありそうだし，特徴を出すためにこういう演出が考えられるようになったんでしょうけど，とって付けたように女の子が出てくるよりは，このテのほうがいいと思いますね。それに短いメッセージの会話が表示されるだけでも，打つ人の表情が想像できて，いいもんです。

熱中度▶▶▶▶▷▷▷　(て)

X1turbo用　5″2D版3枚組 6,800円(税別)
(2ドライブ専用)
マイクロネット　☎011(561)1370

スターシップランデブー

正体不明の宇宙船のなかでのシューティングゲーム，と思いきや，スペシャルステージがメインなんでしょうね，これはきっと。

▶パソコンゲームのあるところ必ずありの，あの手のヤツだ。各面は前半のアクションシーンと，後半のビジュアルシーンに分かれている。前半は，4方向スクロールのアクションシーン。ここでは，敵ロボットを避けながら，各種アイテムを集めることになる。アイテムをたくさん取っておくと，後半で有利になるのだが，どうやら，この面はプレイヤーをじらすためにあるらしい。で，どこかにいる女性を捕まえると，次のビジュアルシーンが待っている。

こっちでは，道具やアイテムを使って実写取り込みのお姉さんに対し，いろいろなことをするわけだ。古くは，マカダムや聖女伝説と同じ手法である。うまくいけば，徐々に脱いでいくのは定石どおり。2パターンのアニメが恥ずかしい。

なお，モデルのお姉さんは，ちょっとケバい水商売風だったりするので，ハーフロ

リータ志向でポップレモンあたりがちょうどいいという人には向かないだろう。

熱中度▶▶▶▶▷▷▷　(お)

▶いつかしら，こんなゲームが出るんじゃないかしら，と思っていた。X68000が出たとき，必ずこのテのゲームはたくさん出るぞ，とみんな思っていた。そして，このテのゲームは誰がどう作ってもB級になるぞ，と僕らは思っていた。案の定そうだった。たとえ，サンプリングでもだえようとも，デジタイズで取り込もうとも，B級になってしまうのは宿命ともいえよう。しかし，B級猟奇SFだって，敵の女の子を××するためにアイテムを集めることだって恥じることはない。デジタイズの絵を相手に“マカダム”しようとも恥じることはないのだ。ただ，スキャップが恥ずべきことは，モデルの人選を誤ったことだ。パソコンの画面に出てくる女の子は，実在の女の子とは違うのだ。このへんの人選はゲーセン版の麻雀ゲームが究めているから参考にするといいだろう。女の子が可愛くないと，ほかがどんなによくできていてもすべてがパアなのであった。

熱中度▶▶▶▶▶▶▷　(K)

X68000用　5″2HD版 2 枚組 8,800円(税別)
スキャップトラスト　☎03(407)0687

ファンタジーゾーン

巨大要塞建設の野望を打ち砕くため，戦い続けるオパオパ。このバランスは，やはり名作と呼べるゲームといえそうです。

▶パステルカラーの美しい画面，軽快なBGM。そして可愛いキャラクターがポンポン飛び回るとっても楽しいシューティングゲームの登場です。

このファンタジーゾーンは，いわずと知れたアーケード版からの移植ですが，できのよさは，あの電波新聞社がやっているんですから保証付き。素晴らしいできになっています。最初のうちは，オリジナルが数年前に作られたゲームだけに，少々時代遅れに感じるのではないかと思っていたのですが，いざゲームを始めてみると，スピーディなゲーム展開と，豊富なバリエーションが用意されたボスキャラなど，古さなんかはこれっぽっちも感じさせないほど楽しくプレイできました。

難易度のレベルは中くらいってところでしょうか。難しすぎず，かといって簡単でもない。非常にプレイしやすいシューティングゲームといえるでしょう。さり気なくお勧めですよ。

熱中度▶▶▶▶▶▶▷　(純)

▶はぁはぁ，ファンタジーゾーンが完成したんですって！　それは凄い。よし，スタートだ。うぉ，音楽は相変わらず完璧だなぁ，電波の移植は。うっ，下に曲名が表示されたぞ，まっ，まさかぁー，X68000のファンタジーゾーンは国内では幻とされていた海外バージョンかぁ。これはうーまーいーぞー。敵の出現パターンも，動きのアルゴリズムもアーケードそっくり。ゲーム中のディスクアクセスは一切ないし，もうダメ。あああぁー，ファンタジーゾーンの世界

に引きずり込まれていくぅー。あっ，あんなところに登録キーが。怪しいぞ，押してやれ。うぁー，得体の知れないメニューが出たぞ。このメニューでは，音楽をオリジナルとアレンジをビシバシ切り換えたり，自分のプレイデータをディスクにセーブできたりするのか。ランクの設定はもちろん，ノーミスのパーフェクトプレイも見られちゃう。これはX68000ユーザー必携の機能満載ゲームソフトの登場だ。ブラボー！

熱中度▶▶▶▶▶▶▷　(善)

X68000用　5″2HD版 7,800円(税別)
電波新聞社　☎03(445)6111

このジャンルは期待度100%

また最近，私は「第4のユニット3」をプレイしてしまいました（というより，すでに敵の弱点をメモっておいたので，どちらかといえば同じビデオを2度見たって感じなんですけどね)。皆さんは，もうプレイしましたか？　今度，X1 turbo版も発売されたので，ネタばらしにならない程度にお話しますけど，私は改めて絶妙なキャラクター設定に感動してしまいました。

まず，なんといっても主人公ブロンウィンとその最大の敵であるダルジィ。この2人の関係と，強力な切札，サイコブースターがダルジィの手の内にあるというストーリーが，この物語を盛り上げています。

それからたぶん，ブロンウィンがほのかな好意を抱いていたユウスケと山田記者。でもブロンウィンは自分がBSであることを，どうしても2人に明かせない。そして，ブロンウィン自身が自分自身のことをよく知らない。これらの絡みが本当によくできています。これだけ基本シナリオがしっかりしていれば，10話でも20話でも続けられそうです。このように，従来のAVGに，テレビアニメ的要素を加えた軽くてしっかりしたシナリオのAVGの今後も期待したいものです。　(で)

THE SOFTOUCH

●ジェノサイド

妥協を許さぬハードアクション

Ogikubo Kei
荻窪 圭

アクション派ゲーマーたちが，発売前から心待ちにしていたこの「ジェノサイド」。とにかくデカイキャラが画面狭しとビュンビュン飛び回る。いずれにしても，X68000のオリジナルゲームの世界は，ますますハードでスピーディな展開を見せ始めたようです。

X68000用　5"2HD版4枚組　8,800円(税別)
ズーム　☎011(613)0191

序章：これが噂のジェノサイド

なにげなくジェノサイドをやっていて，なにか引っかかるものを感じた。この違和感はなんだ？　よく画面を見る。そうなのだ。画面には戦闘フィールドの他には自分のエネルギーしか表示されていないのだ。そう，ハイスコアどころかスコアさえもない。表示されないどころか，スコアなんて概念がどこにもないのである。妥協を許さない，戦い続けることのみを義務付けられた男のゲーム。これがジェノサイドの正体だったのである。

＊　　　＊

MESIAとかいう“超ニューロコンピュータ”(笑)を壊し，たとえ砂漠のなか放射能のなか，ひたすら前進突き進め，というバトルゲーム。ずんぐりしたトレーサー（なかに乗っている人の動きをトレースして動くのだそうだ。こいつに乗ったら，うっかり鼻もほじれやしない），右手にはサーベル。たったひとりで砂漠を前にし，途方にくれる。トレーサーのコンピュータはよく喋る。ランディというのだそうだ(笑)。こんなペチャクチャと喋るコンピュータを作るヤツのセンスを疑う。

第1ステージである砂漠では，ノリのいいBGMがステレオから響く。シュワッと，地中からミドリのヘビが丸まって飛び出てくる。敵が出なくなると，先へと進む。源平の剣と違ってライトサーベルだから，硬いものを叩いたからといって曲がったり切れ味が鈍ったりしない。しかし，こっちには飛び道具はないくせに，敵さんのロボットなんかはバコバコ弾を撃ってくるんだ。これは卑怯だ。特に緑色の警備ロボット野郎は困ったことに，離れれば弾を撃ってくるし，逃げれば飛びかかってくるのだ。もうこれは接近戦と覚悟を決め，上段からサーベルを振り下ろすしかない。

実践講座1：ジャンプを極める

楽して実を取る実践講座である。ジャンプはBボタンによってなされる。ジャンプを制する者はマンガ界を制す，といわれるようにジャンプを使いこなすのは重要である。ジャンプしたあとの姿勢制御は次の3つに集約される。

1）ジャンプした方向のままで前や後ろへ行く。
2）ジャンプして後ろを向いたりする。
3）ジャンプして空中で伸身宙返り半ひねり。

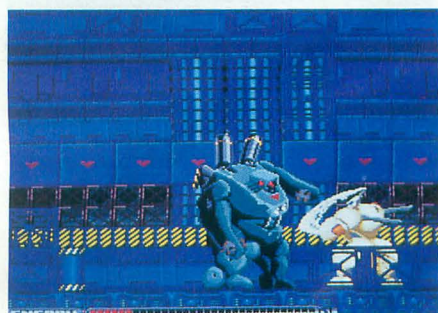
である。これらは曲者である。ジャンプキーを押したままにするかすぐ離すか，アタックキーを押すか押さないかなどで違ったパターンを見せるからだ。

普通にジャンプしてレバーを逆方向へ入れると，後ろを向くが，空中でサーベルを振り下ろしたままレバーを逆に入れると，進行方向を向いたまま後ろへ下がる。空中でサーベルを振り下ろすにはジャンプボタンは離してなければならない。さらに伸身宙返り半ひねりはジャンプをして，ジャンプボタンを離し，ジョイスティックレバーを上に入れるとできるのだが，この上というのがミソである。

人間行動心理学的にいって，ジャンプボタンを押したとき，つい高く跳びたいと思うあまり，無意識のうちにレバーは上を向いてしまうのだ。しかし，それでも伸身宙返り半ひねりはしない普通のジャンプだ。なぜなら，行動心理学的にいって（本当かなあ），長くジャンプしようと思うあまり，無意識のうちについジャンプボタンを押し続けてしまうからである。

つまり，だ。興奮した局面では，空中でサーベルを振りたい！　と，思うあまりジャンプボタンを離してアタックボタンを押そうとした途端，無意識のうちにレバーが上に入っていたせいで伸身宙返り半ひねりをして無防備にも敵に背中を見せて着地して殴られたとか，伸身宙返り半ひねりで敵の後ろに回ろうと必死にレバーを上に入れるけれど，ジャンプボタンの指が膠着して離れず，ただ敵に突っ込んだだけになったりするのである。サーベルを振り下ろそうと必死にボタンを押すのに，ジャンプボタンを押しっ放しにしていたおかげで敵に激突することも珍しくない。このくらいジャンプを制するのは難しいのだ。

＊　　　＊

そんなこんなで，1面のボス“力持ちの工作機械野郎”に出会ってしまった。グアッと叫びながら長い腕で殴りかかるのはまだしも，真上に放ったミサイルが画面を飛び出てから落ちてくるとか，離れると自動

こいつがガコガコと迫り来る工作機械野郎

追尾ミサイルを撃ってくるなど実に猿山のボスに相応しい野郎なのだ。私みたいに，「グアッといって殴りかかる姿が魅力的！」などと倒錯する奴以外はさっさとやっつけてしまおう。サービスか，マニュアルにこいつのやっつけ方が書いてある。ふんふん（当然，ESCキーでポーズしてから読んでいるのよ)，ナニッ！　伸身宙返り半ひねりの間は敵弾の当たり判定をしていないだと。

なに斬ってんだって？　こいつが噂の巨大戦艦

かに道楽の看板じゃあるまいし，とにかく硬い

実践講座2:効率のよい斬り方

なんと，上段で振り下ろすよりも，連射で叩くよりも，よくきく技があったのだ。ジャンプしてサーベルを振り，そのまま振り下ろした姿勢で敵に突っ込む（アタックボタンを押したままにするのだ！)。その名も「下段のジャンプ突き！」である。しかも，空中にいる間は，下段のジャンプ突きの姿勢のままで前や後ろに(少しだけれど)動けるのだ。これで，素早い敵にも対処できる。もちろん，着地したら構えの姿勢に戻ってしまうので注意である。この技と，ジャンプバリエーションを使いこなせば，ジェノサイド極めの道（ハマりの道ともいう）に一歩近づいたといえよう。

＊　　　＊

さもありなん。廃虚を通りすぎようとすると，ズゴゴゴゴとどんな鈍い人間でもすぐわかる嫌な予感の音。空中から画面3つや4つは優にありそうな巨大戦艦。しかも，丸いBB弾みたいな弾をダバダバと撃ってくるかと思えば，ハッチが開いて猿を改造したような小さくて，自分の何倍もすばしっこいちょこまか忍者野郎がどどどっと飛び降りて来るのだ。

それでもって，巨大戦艦を追い返して建物のなかを進むと，巨大な貨物運び用エレベーターがあって，このなかでしゃがんで待ってよう。そして地下を抜け出すと，いきなり洞窟。飛んでくるゴキブリを避けたあと（伸身宙返り半ひねりで簡単に避けられるが，別に戦ってもいい)，ゴゴゴゴとドリル付きの恐怖の“天安門戦車”が出てくる。これは，何度も踏み潰されるうちに「この，天安門戦車が！」と叫んだのが名前の由来だ。

こいつが硬いわ強いわの困ったもので，しかも，ただの下段のジャンプ突きでは壊せない。滅多にこんなことはしないけれど，壊し方を教えてやろう。ただ，人の言葉に頼らず，敵を見て瞬間にその弱点を看破する目もジェノサイドには必要だ。

この天安門戦車は前方下部のドリルを壊す。すると，BB弾みたいな弾を乱射してくる。が，ビビって下がってはいけない。強力なレーザー砲を撃ってくるからだ。密着している限り，レーザー砲は大丈夫。ドリルを壊したら，次は前の赤い窓を壊すのだ。これにはより深い踏み込みが必要なので，留意すること。ここで，焦るあまり行動心理学の罠に落ちると，伸身宙返り半ひねりして敵の後ろに回ってしまい，バックしてきた戦車に礫かれてしまうという悲惨な目にあいかねない。

で，前の赤い窓を壊したら，次は後ろの赤い窓だ。伸身宙返り半ひねりで後ろにまわり，下段のジャンプ突きでKOだ！

と，簡単に書いているけど，苦労した。

実践講座3:正しいベティの使い方

この守護霊君は，アタックボタンを押し続けると（おっと忘れてはならない連射解除)，どんどん霊気を溜めて白く光っていく。ここで，パッと離すと，そのときレバーの入っていた方向へヒューと飛んでいき，霊気を放出してヨーヨーのように戻ってくるのだ。やったぜ強力飛び道具！　といいたいところだが，これもまた使いこなすのが難しい。最初にマスターすべきは，斜め45度上方飛ばしと，斜め45度下方叩きつけだ。そうすれば，飛んでいるヤツにも背の低いヤツにも有効。

敵の攻撃を一度でも受けると充電し直しだから，敵が密集しているところや，霊気を溜める間を与えてくれない高速移動野郎には向かないので気をつけること。

＊　　　＊

3面はとにかく，不気味突然変異メタモルフォセス野郎の束，メタモルフォセス群島だ。こんなところ，さっさとおさらばしたい。たとえば，“ふわふわした岩かと思って切るとなかから生まれたバルカン砲野郎”が相手なら，守護霊を2回くらいぶつけると生まれる前に破壊できる。

で，この面のボスキャラである巨大（本当に巨大）戦艦となると，もう，「ごめん，俺が悪かった。許して。謝るからなかに入れて！」という世界だ。

とにかく，浮いているのを下に降ろして止めないと話にならない。それには，下部の浮力エンジンを壊す。で，だんだん下がってこられると，こちらも下に潜り込めなくなるから，下を潜って前に回りながら壊していくのがベスト！（並の根性じゃできないんだ，これが)。それでもって，エンジンは前方についているから，それを壊して動きを止める。それでもって，戦艦の上へ出る。そこで，怪しいところが数カ所あるので，そこを壊せば，戦艦のなかに潜り込めるというわけ。

なかに入ってしまえば，あとは根性あるのみ。不気味な顔だけど，頑張って壊そう。

＊　　　＊

残りページがもうないから，急ぐぞ。4面は放射能メタモルフォセスバイオ野郎の溜り場だ。叩くたびに胞子をまき散らしてメタモルフォーゼする野郎とか，クラゲ野郎とかぷわぷわ浮く卵だけれど壊すとなかからベロ出しゲロゲロ野郎とかいろいろと楽しい，ほんっとに楽しい。あー楽しい。硬いヤツも嫌だけど柳のようにしなやかな，暖簾に腕押しキャラもやっかいだ。

ここのボスはなんと，ズームのある北海道名産のカニだあ。腹と背中に目があるので，下の目はカニが跳んだ隙に真下から守護霊をブチ込めばなんとかなる。問題は上の目だ。これは，画面両サイドの棚に飛び乗れるから，まず飛び乗って，泡をかわしながら機会をうかがい，えいやっとカニの背中に飛び乗って，叩け，叩け，叩けー。はあ，はあ。いっとくけど，ジェノサイドにはセーブ機能はないぞ。コンティニューはあるけれど。はあ，はあ。

次が最終面である。内容は内緒である。ただ，最後に顔を3つ持つボボブな野郎が飛んでいる，とだけいっておこう。

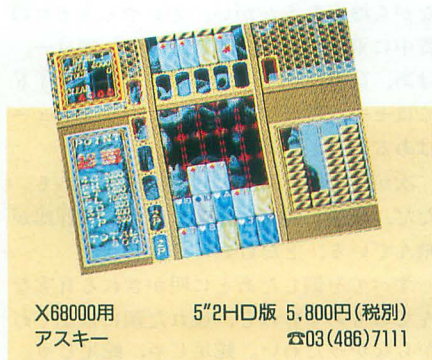
すべてを倒したあとに明かされる真実なんてのもあるけれど，疲れた頭にはよくわからなかったわい。蛇足じゃ，蛇足じゃ。

THE SOFTOUCH

●琉球

シンプルさのなかに伝統の味を見た

Kunitsu Yoshio
国津 良男

久々に心地よくハマれるパズルゲームが登場した。なんとこのゲームは，読者からの投稿作品であるという。さらに驚くのは，これの送られた先が某LOGiN誌であるというのだ。というわけで，Oh!Xでも投稿をお待ちしています(なんなんだ，これは)。

X68000用　5"2HD版 5,800円(税別)
アスキー　☎03(486)7111

新・ハマりゲームの登場

「上海」そして「TETRIS」(注1) にハマった皆さん。おめでとう。あなたの大切な時間を，際限なく食い潰してしまう，プレイヤーわくわく，お母さんピリピリのゲームが，またまた登場してきた。その名も「琉球」。琉球とは，もちろん沖縄のことである(注2)。

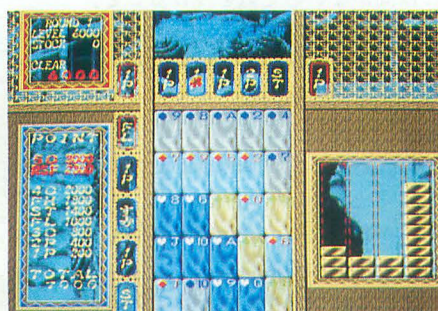
これはいったい，どんなゲームかっていうと，早い話が2次元ポーカーだったりするのだ。トランプは，西洋カルタともいうぐらいだから(注3)，沖縄特産でもなんでもないのはもちろんのことだ。タイトルが「琉球」なのは，BGM や背景のグラフィックなどの演出部分が沖縄風味になっているからであって，決してどこかの編集部が撮影取材と称して沖縄に遊びに行きたいがために付けられた名前ではない。本質はあくまでも純粋なトランプゲームである。

それでは，その2次元変則ポーカーのルールを説明しよう。

ルールの説明

まず，ゲーム画面の写真を見てほしい。真ん中上には，4枚のカードが表になっていると思う。このなかから1枚，好きなカードを取り，下のフィールドのどこかに落とす。そんでもってまた，上から1枚取り，下のフィールドのどこかに落とす。とまあ，これを繰り返すわけだな。

下のフィールドには5×5枚のカードが置けるスペースがある。ポーカーってのは，5枚でひとつの役を作るのだからして，2次元ポーカーの場合は，5×5枚並べるってぇのは，当たり前田のクラッカーである(注4)。

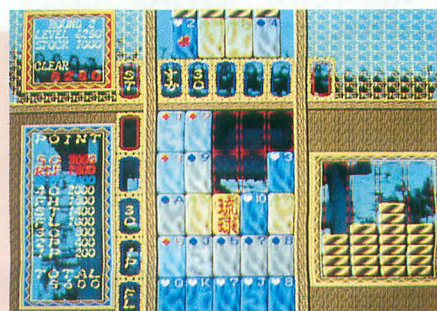
さて，下のフィールドに25枚のカードを敷き詰め終えたら，ひとまずゲームは終了だ。そのとき，縦，横，斜めの役の合計が，クリアポイントを越えていれば次の面に進めるが，足りなければその場でゲームオーバーということになる（コンティニューモードはあるけど)。もちろんクリアポイントは，面が進むにつれて高くなってゆく。ワンペア，ツーペア，スリーカード(注5)，はい出来上がり，と簡単にいくほど世の中そんなに甘くない。

役は，上がファイブカードの3000点やロイヤルストレートフラッシュ2800点から下はワンペアの200点，って決まっている。ワンペアってなに？　って人は，いまさらいないよね。ま，遊び方は，だいたいこんなところかな。

のび太君がんばる

さて，デモプレイを，ドラちゃんとのび太君にお願いしよう(注6)。

の：ねえ，ドラえもん。うざったいルールの説明はもういいからさあ，早く遊ぼうよ。
ド：うん。じゃあ，のび太君，先にやってもいいよ。ボクはドラやきを……。
の：わーい。えーっと，まず上のカードを見てぇ，4枚とも全然揃ってないや。だから，どれを取っても同じだから，ハートの7を取ってポトン！　と。
ド：バカだなあ，のび太君は。7を左隅に置いちゃって。スペードのAがあるじゃないか。これを隅に置けば，3方向ロイヤルストレートフラッシュが狙えるのに。
の：そっか。えーっと，じゃあ，スペードのAを取ってぇ，ポトン。
ド：バカだなあ，のび太君は。もう，左隅に7を置いちゃったんだから，わざわざ右隅にAを置く価値はないってのに。それに，

注1） ときどきTETЯISと書いてあったりするけど，あれは装飾。"Я"は「ヤー」と読むのだし，だいたいロシア語にはIとかSなんて文字はない。JIS漢字コードの2721Hからを見てほしい。
注2） 本来は，中国による沖縄の呼び名だそうな。
注3） と思ったら，トランプ（カード）の起こりには，中国起源説，インド起源説，エジプト起源説などがある，と小学館の万有百科大事典に書いてあった。
注4） ちょっと古すぎたかしらん，このネタは。
注5） 石野真子の歌。えっ？　知らないって。
注6） ©藤子・F・不二雄/小学館

縦，横，斜めの役が勝負のポイント

琉球カードはやはり中央か上の両隅に置きたい

よく見てごらんよ。さっき取った7の後ろから，琉球カードが現れたでしょ。これを右隅か真ん中に置くことを考えたほうが，よりハイスコアを狙えるよ。
の：ねぇ，この琉球カードってなに？
ド：早い話がジョーカーさ。なんにでもなる便利なカードなんだ。
の：そうなの，ふーん。
ド：それより，もっとよく考えなきゃ。時間制限はないんだから。
の：わかってるよ，うるさいなあ。うーんうーん。
その後10分経過。
ド：のび太君，どう？ 決まった？ のび太君ってばぁ。
の：ぐぅー，ぐぅー。
ド：また寝てる。まったく，寝ることにかけては，天才的なんだから。ほら，起きて！起きて！
の：もう，さっきからいちいちうるさいなあ。そんなにいうんなら，ドラえもんやってみせてよ。
ド：イシシシ，じゃあ見ててよ。この場合は，縦方向にフラッシュを狙ってと，しまった，手がすべった！
の：ドラえもんのその丸い手（注7）じゃ，キーボードを操作するのは無理だよ。マウスを使ってごらんよ。
ド：ギャー，ねずみー！
とまあ，こんなふうに楽しく遊べるゲームなのだ(?)。

沖縄へのいざない

さて，琉球の舞台になった沖縄である。面クリアごとのごほうびは，沖縄の伝統工芸品や，お祭りの風景などの，実写取り込みのグラフィックである。沖縄県は60余りの島々からなるのだが，どうやら，撮影は本島のみで行われたようだ。

旅行好きの友人によると，これは民家の屋根の上なんかにある守神シーサーだ，とか，これはエイサーって祭りの様子だろうとかいちいち教えてくれた。沖縄の海はほんっとにきれいで，スキューバダイビングは病みつきになるぞ，とも話してくれた。

私自身は，歴史や文化に疎く，旅行にもあまり興味を感じないたちだ（ゲゲボツアーには行きたいと思っている）。ダイビングどころか，クロールの息継ぎもできない。しかし，これらのグラフィックから，エキゾチックな，ロマンチックな香りを感じ，なんかいいなあ，と思ってしまう。本当は，行ってみないと，よさなんてわかんないだろうけど，東京から沖縄まで飛行機で片道4万円するらしいし，ハブが出るぞー，なんて脅かされるし，そんな時間もないし。

あと，BGMもいかにも沖縄っぽい曲で，グラフィックと相まって沖縄情緒を前面に押し出している。

ゲームセンターのポーカーゲームにありがちな，ラスベガスのギンギンギラギラムードとは違って，新鮮で，かつ落ち着いたムードを醸し出していて，とっても好感が持てるのだ。

ハマりに関する一考察

琉球はパズルゲームではあるが，そんなに頭を使うものではない。使ってるのか，使ってないのか，よくわからないような，なまぬるさを持っている（注8）。この点が実はパズルゲーム作成におけるポイントなのである。ガチガチに，数学化されていては，一般ウケするわけないし，あまり簡単すぎてもつまらない。ある程度，偶然が作用する思考ゲーム，これだ。

偶然の要素が含まれていないと，人々はついてこない。もし失敗しても，単に運が悪かったのだ，と思わせるだけの逃げ道を，初めっから用意しておいてやらなければならない（注9）。頭が悪い，機転が利かないなんて，ゲーム中に思い知らされてはたまらないではないか。それに偶然は，緊張と期待をもたらすという効果もある。

かといって，偶然がそのほとんどを支配し，テクニックの蓄積もできないようなパズルでは，お話にならない。プレイヤーを虚しい気分にさせてはいけない。場数を踏んでも，初心者と大差なし，ってのでは悔しい。はっきりいおう。経験を積んだ人間は，初心者を見下す特権をほしがるものなのだ（注10）。

この琉球には，EASYやHARDなどのモードが用意されていて，これらの違いは，HARDのほうがより多くのカードが見えているってことにある。つまりは，偶然性が低くなっている分だけ，プレイヤーは一生懸命考えなくっちゃならない，ってわけ。

この，偶然と必然のゲームバランスが非常に大切なわけで，琉球はここんとこ，特にHARDモードはうまくできている。で，私のような者でも気軽に遊べるのだ。「いやー，あっぱれ，あっぱれ，はげひよこ（注11）」。

投稿ソフトなんだって

このソフト，もともとは，LOGiN 誌への投稿作品であり，1989年7月7日号を見てみると，準堀井賞に選ばれている。作者は京都の大学生。商品化の際には，いくらか手直しされるらしいが，個人の作品がパッケージ化されるってのは，いまのご時世ではなかなかに珍しいことだと思う。

面をクリアするとこうした沖縄の風物が見られる

投稿作品では，ソフトハウスの作った商品には歯が立たないというのが相場。がんばって作ったイースもどきや，グラディウスもどきをよく見かけるが，残念ながら完成度がいまひとつなものが多い。だからって，作ってもしょうがないとはいわない。やはりここは，アイデア勝負の作品をぶつけてみるのがいい。

別に長いプログラムでなくても，オリジナリティと，独創的なセンスのあるものがいい。なんて，いうは易し，行うは難しだあね。やっぱり。

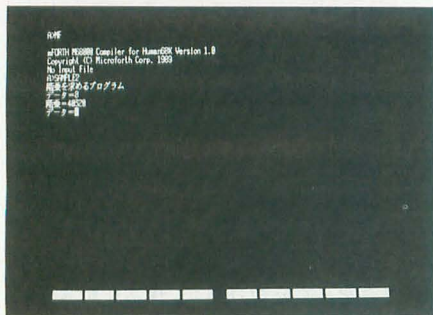
でも，なにか作ったら，LOGiNもいいけど，Oh!Xもよろしくね，なのであった。ついでに，(で)のショートプロぱーてぃも，お忘れなく。

*　　　*

昨日，今日と東京は雷雨。こんな日は，出かけるのもイヤだし，友達を呼ぶのも迷惑ってもんだ。でもこんな日は，プログラミング日和，琉球日和と考えるといいかもね（注12)。とにかく，軽くサラッとハマッて，しかもチョッピリ難しい。この沖縄風味の味付けは，なかなかにおいしゅうございました。

注7）ペタンハンドといって，物がくっつく仕掛けになっている。
注8）ひょっとして，あまり考えずに遊んでいるのは私だけだろうか。中学時代，国語の先生に，あなたはなにも考えてないんじゃないの？といわれてしまった，苦い記憶が甦る。
注9）受験に失敗したとしても，運のせいにできる受験制度で幸せ。もし知徳体力測定機なんてもので(できるわけないけど)，人間が正確に順序付けられたとしたら滅入ってしまう。いい訳の余地がない人生ってぇのはつらい。
注10）年長者を尊敬しようっていうのも，これに通じるんじゃないかな。
注11）友人の口癖。意味不明。
注12）といいつつ，コミケカタログのチェックに忙しい昨今。お勧めサークルがあったら教えてください。

THE SOFTOUCH

●mFORTH Compiler

静寂を破って久々にFORTH登場

Nakamori Akira
中森　章

自己増殖型という独得のスタイルを持つFORTH。最近では少し疎遠になった感はあるものの，実際に使ってみるとなかなかに愛着の持てる言語なのです。それでは，X68000に発売されたばかりのmFORTHの特長をこれからご紹介することにしましょう。

X68000用　5"2HD版　19,800円(税別)
マイクロフォース　☎03(756)1988

その昔(といっても5年ほど前)，本誌に「ふぉーす3部作」を発表した清水和人氏は，FORTHを評してこう述べています。「FORTH は誰も知らない自分だけの言語を作ることのできる万能の言語，あるいは一子相伝の必殺の言語」なのだと。このように，FORTHという言語はそれに触れた者を虜にしてしまう，魔性を秘めた言語なのです。

このFORTHは，アメリカの天文台に勤めていたチャールズ・ムーアという人が電波望遠鏡を操作するために開発した言語だといわれています。その後ムーア氏は天文台を辞め，自分で会社を設立してFORTHの普及活動に努めたといいますから，FORTH の素晴らしさをひとりでも多くの人に知ってもらいたかったのでしょう。そのような事情もあって一部には浸透していたFORTHでしたが，Oh!MZの誌上でS-OSの「magiFORTH」が発表されて以後，次第にFORTHの名前を聞く機会がなくなったようです。

しかし，その静寂を破るように，このたびX68000用に伝説のFORTHが発売されることになりました。それも従来のようなインタプリタではなく完全なコンパイラとしての登場です。かつて清水氏が絶賛した必殺の言語に触れるチャンスがX68000ユーザーにもやってきたのです。

FORTHという言語の構造

新しい読者の皆さんのなかには，FORTHという言語を知らない人も多いと思います。そこで，mFORTHの説明に入る前にFORTHの一般的な特徴を示しておきます。

1) ワードと自己増殖性

FORTHはちょっと変わったプログラミング言語です。FORTHの命令はワードと呼ばれ，ワードを定義することがプログラミング，ワードを実行することがプログラムの実行になります。新たなワードの定義は既存のワードを組み合わせて行われます。そして，一度定義したワードはあらかじめシステムに備わっているワードとなんら区別なく使用できるようになります。つまり命令が増えたことになるわけですね。

また，FORTHのシステムはほとんどの部分がFORTH自身で記述されています。このためちょっと知識があればシステムのなかで気に入らない箇所を書き換えることも容易です。これはシステムが使用するワードを自分が定義したワードで置き換えることです。このようにFORTHは使い始めた日からどんどん成長していく自己増殖型の言語なのです。

2) スタックと逆ポーランド記法

スタックとは，一般のCPUで関数をコールするときに戻りアドレスを入れ，関数からリターンするときに戻りアドレスを取り出すために使われる，いつものあれです。つまり，最初に入れたデータを最後に取り出す構造を持ったデータの保存場所なのです。

FORTHでは関数コール用の戻りアドレスを入れるスタック（リターンスタックという）と独立にデータの演算を行うスタック（パラメータスタックという）を持っています。また，ループ構造を効率よく実現するためのスタック(DOスタックという)を持つFORTHもあります。

さて，このスタックはFORTHを特徴づける最大の要因といっても過言ではありません。すべてのワードはスタックに対してなんらかの操作を行います。たとえば，

12　23　+

というFORTHのプログラムは次のように動作します。まず“12”というワードによって12という値がスタックに入れられ，次に“23”というワードによって23という値がスタックに入れられます。そして最後に“+”というワードによってスタックの先頭(23)と2番目(12)のデータが取り出され，加算されたその結果(35)が再びスタックに入れられます。このようにワード（演算）の実行結果はスタックの先頭に残ります。また，演算にスタックを使用する性格上，ワードの並びは逆ポーランド記法になります。

3) ボトムアップなプログラム開発

多くのFORTHの処理系では，入力したワードの定義をその場でコンパイルして新しいワードを作ります。このため，ワードの定義のなかで用いるワードはすでに定義されているものでなくてはなりません。これをプログラムのモジュール化という観点から見ると，あらかじめ必要となるワード(サブルーチン)をすべて作っておいてから，最後にそれらのワードを集めて目的となるワード（メインプログラム）を作るというボトムアップなアプローチになります。いわゆる構造化プログラミングでは最終的な目的を次々と個々の機能に分解していくトップダウンなアプローチがなされますが，よほど大規模なプログラムを作るのでない限り，ボトムアップのほうが早くプログラム開発できるものなのです。

4) ROM化

FORTHは電波望遠鏡の制御用に開発されただけあって，プログラムをROMに焼

き込んで制御機器などに組み込むことを最終目的としています。このためFORTHのコンパイラは非常にコンパクトなオブジェクトコードを出力するように作られています(コードサイズが小さいほうがROMに焼く時間を節約できる)。このためか、一時期FORTHはシングルチップマイコンの開発言語として注目を浴びていたようです。

mFORTHの特徴

最近は主流ではなくなったからといっても，FORTHが扱いにくい言語だということはありません。先に述べたようにFORTHはプログラム開発はしやすいし，拡張性も優れています。ワードだけからなり，データの受け渡しに必ずスタックを使用するというシンプルな言語体系は，美しささえ感じさせてくれます。

また，あとから述べるように，FORTHはアセンブリ言語とのリンクも容易にできます。ですからFORTHの愛好家がいなくなることはまずないでしょう。しかし，彼らがFORTHの処理系を身近なパソコンに求めるとき，それがX68000でないとしたらそれは悲しいことではありませんか。X68000でFORTHが動く。それだけを考えても，今回のmFORTHの登場は大きな意味があるといえるでしょう。

それでは，mFORTHの特徴について順に見ていくことにしましょう。

1) コンパイラ方式と構造化プログラミング

mFORTHはインタプリタが主流の従来のFORTHと違い完全なコンパイラです。インタプリタでないため対話性には欠けますが，3パスのコンパイラとしてファイルをすべて読み込んでから，オブジェクトコードを出力するため，従来の使用するワードが先に定義されていなければならないという制限はありません。これで，プログラム開発において，従来ではボトムアップなアプローチしか許されなかったFORTHでも，構造化プログラミングに適するトップダウンなアプローチもできるようになったのです。

やはり，構造化プログラミングができるかどうかによって一般ユーザーへの受けは大きく変わってきますから，この点は重要ですね。また，プログラムの構造化を助けるために“SWITCH”や“CASE”などといったほかのFORTHでは見られないような便利な制御構造も備えています。

2) アセンブリ言語の記述とマクロワード

FORTHのワードは既存のワードを組み合わせて作るのが建て前ですが，mFORTHではアセンブリ言語で直接ワードを定義することもできます。その場合，普通のアセンブラと同様のニーモニックを使用できるので比較的簡単にワードの定義ができてしまいます。アセンブリ言語が使えればもう怖いものはありません。アセンブリ言語とリンクを行うことでCPUやパソコンのすべての機能を利用することが可能になります。

mFORTHはまさに万能の言語に成長する可能性を持っているのです。

3) 分割コンパイラとライブラリ

mFORTHではプログラムの分割コンパイルができます。これは一般の分割コンパイルの概念とは少し異なっていて，あるファイルをコンパイルするときには別のひとつのオブジェクトファイル（コンパイル済みのファイル）だけを参照することができます(これをリンクという)。オブジェクトレベルでファイルをインクルードするとでもいえばよいのでしょうか。

しかし，リンクするファイルがさらに別のファイルをリンクしている場合は，そのファイル内で定義されたワードもさかのぼって継承することができるのです。一般の分割コンパイルのイメージは，単独で作成したプログラムを後からつなぎ合わせることですが，mFORTHの分割コンパイルはライブラリ(あるいはmFORTHそのもの)を順次拡張しながら組み込んでいくというものでしょう(図1)。そして，mFORTHの基本的なワードはこのようなオブジェクトファイルの形式で供給されていて，それをリンクすることでいろいろな基本操作ができるようになっているのです。参考までに，図2にmFORTHを用いてプログラム開発を行う場合の処理手順を示しておきます。

4) X68000の専用機能をサポート

X68000の言語であるからには，X68000の機能をすべて扱えるものであってほしいものです。その点，このmFORTHは抜かりありません。もっとも，アセンブリ言語とのリンクが容易にできるのですから，自分でコツコツとX68000専用の機能(グラフィック，スプライトやFM音源)を扱うワードを作ることもできます。

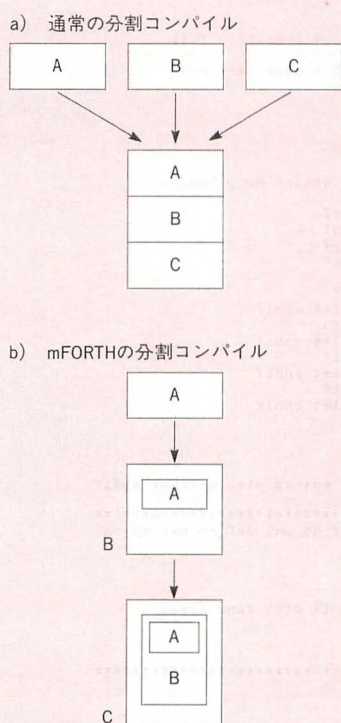
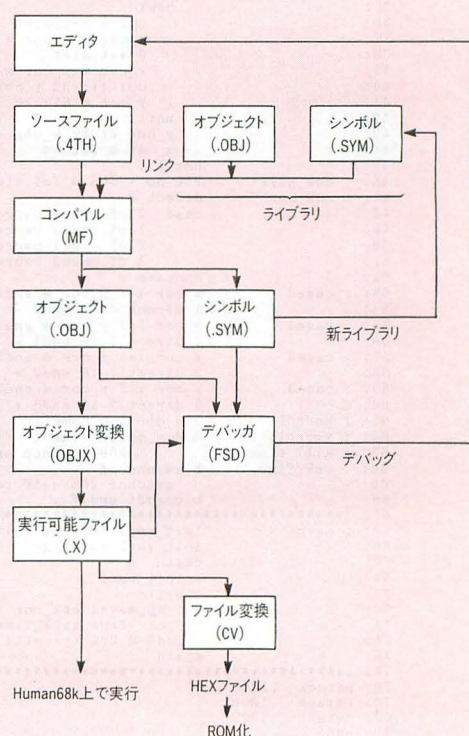
しかし，そんな努力をしなくてもmFORTHではHuman68kのDOSコールとX68000のIOCSコールをサポートするためのライブラリが付属しています。これらのライブラリはソースファイルの形式で供給されていますからワードの改造や拡張も容易にできるでしょう。

5) シンボリックデバッガを標準装備

mFORTHはインタプリタではないので，対話性の悪さが気になるところです。インタプリタであれば，1つひとつのワードをデバッグしながらプログラムを作っていけ

図1　分割コンパイルのイメージ

図2　mFORTHの開発手順

るので完成したプログラムにバグはまずありません。しかし，コンパイラではそうもいきません。多くの場合，定義したワードにバグがあるかどうかは実行してみるまでわかりません。実行してみてうまく動かないときは，どのワードが悪いのかと考えを巡らすのです。mFORTHではこのような状況に対処できるようにシンボリックデバッガ“FSD”が標準装備されています。

このデバッガを使えばプログラムで定義したワード名をデバッグするプログラムのアドレス代わりに指定できるので便利です。この点FSDは，Cコンパイラパッケージの DB.Xよりも強力なデバッガといえるかもしれません。もっとも，FORTHのプログラムはスタックポインタを勝手に書き換えてしまうので，DB.XではFORTHのプログラムをトレースすることすらできません。

6） ROM化に最適

FORTHは制御用を意図した言語であり，ROM化に向いた言語であることはすでに述べました。mFORTHではROM化という観点でもいろいろ便利な機能を有しています。疑似命令でプログラムのコード部(ROMに置かれる）とデータ部(RAMに置かれる）のアドレスやスタックポインタの値を自由に設定できるようになっています。また，制御装置に組み込んだプログラムをICEでシンボリックデバッグするためのシンボルファイルを出力するツールも用意されています。これほどまでにROM化を意識したmFORTHですが，X68000用のバージョンにはオブジェクトプログラムをHEXファイル(ROMライタの入力形式）に変換するツールがありません。Human68kのCV.Xを使えばよいと考えたのかもしれませんが，これは少しマイナスポイントです。

リスト1

```
==================  forth.4th  ==================
    1:          flink   iocslib
    2: ;**********************************************
    3: ;*  これは mFORTH のサンプルプログラムです       *
    4: ;*  -----------------------------------       *
    5: ;*    スペースで終了、それ以外のキーでポーズ。  *
    6: ;*    ポーズの解除はなにかキーを押す。          *
    7: ;*                                            *
    8: ;*                     中森  章 (7/23/1989)    *
    9: ;**********************************************
   10: var   col    x_cor  y_cor  direct times
   11: var   end0   end1   end2   end3
   12: ldata PAT = {   00000000h, 00000000h, 00000000h, 00000001h,
   13:                 00000012h, 00000123h, 00001234h, 00012344h,
   14:                 00012345h, 00123456h, 00123456h, 01234567h,
   15:                 01234567h, 01234567h, 12345678h, 12345678h,
   16:                 00000011h, 00011122h, 01122233h, 12233344h,
   17:                 23344455h, 34455566h, 45566677h, 56677788h,
   18:                 67788899h, 788999aah, 789aaabbh, 89aabbcch,
   19:                 89abccddh, 89abcdeeh, 9abcdeefh, 9abcdef0h }
   20: : init_var1     0fffh col ! .
   21: : init_var2     x_cor 0! y_cor 0! col 0! .
   22: : init_var3      16 x_cor !   16 y_cor !  0 direct !  0 times  !
   23:                 496 end0  !  496 end1  !  0 end2   !  1 end3   ! .
   24: : init_gr       8 crtmod g_clr_on b_curoff .
   25: : reset_gr      16 crtmod .
   26: : init_sp       sp_init drop 256 do i sp_cgclr enddo .
   27: : inc_col       col @ 1000h + dup col ! .
   28: : def_sp_pal    15 do i 1+ 1 inc_col spalet drop enddo .
   29: : def_sp_pat    0 PAT 1 sp_defcg .
   30: : sp_move4      80000000h x_cor @      y_cor @       0100h 3 sp_regst
   31:                 80000001h x_cor @ 16 + y_cor @       4100h 3 sp_regst
   32:                 80000002h x_cor @      y_cor @ 16 +  8100h 3 sp_regst
   33:                 80000003h x_cor @ 16 + y_cor @ 16 + 0c100h 3 sp_regst .
   34: : setup_gr      begin
   35:                    begin
   36:                       col @ setpalet
   37:                       x_cor @ y_cor @ over 32 + over 32 + fill
   38:                       0 setpalet
   39:                       x_cor @ y_cor @ over 32 + over 32 + box
   40:                       col 1+! 32 y_cor +!
   41:                       y_cor @ 512 >=
   42:                    until
   43:                    y_cor 0! 32 x_cor +!
   44:                    x_cor @ 512 >=
   45:                 until .
   46: : chg_palet     256 do i dup 8 fsl times @ + gpalet drop %enddo .
   47: : chk_cor       direct @
   48:                 case  0 of case0 hscroll endof
   49:                       1 of case1 vscroll endof
   50:                       2 of case2 hscroll endof
   51:                       3 of case3 vscroll endof
   52:                 endcase .
   53: : case0         x_cor 1+! x_cor @ end0 @ > if
   54:                 1 direct ! 16 end0 -! chg_palet endif .
   55: : case1         y_cor 1+! y_cor @ end1 @ > if
   56:                 2 direct ! 16 end1 -! chg_palet endif .
   57: : case2         x_cor 1-! x_cor @ end2 @ <= if
   58:                 3 direct ! 16 end2 +! chg_palet endif .
   59: : case3         y_cor 1-! y_cor @ end3 @ <= if
   60:                 0 direct ! 16 end3 +! chg_palet endif .
   61: : hscroll       0 x_cor @ 0 home .
   62: : vscroll       0 0 y_cor @ home .
   63: : kill_time          200 do fnop enddo .
   64: : chk_fine      b_keysns if
   65:                    getchar 20h = if reset_gr system else getchar endif
   66:                 b_curoff endif .
   67: ;***********************************************************************
   68: : main          init_var1 init_gr init_sp def_sp_pal def_sp_pat sp_on
   69:                 init_var2 setup_gr
   70:                 begin
   71:                   init_var3
   72:                   begin
   73:                     sp_move4 chk_cor times @ 1+ 0ffh fand times !
   74:                     chk_fine kill_time
   75:                   end0 @ 256 <= until
   76:                 again .
   77: ;***********************************************************************
   78: pstack  ,100h
   79: rstack  ,100h
   80: dstack  ,100h
   81: end
```

図3　サンプル実行画面

サンプルプログラムの説明

これまで，mFORTHについて説明してきましたが，言葉だけではFORTHがどのような言語かはっきりしないのでサンプルプログラムを作ってみることにしましょう。リスト1がこのmFORTHで記述したサンプルプログラムの例です。このプログラムは本誌1989年2月号でスプライトを扱うアセンブリ言語のプログラム例(87ページ)として発表したものに，スクロールするグラフィック画面の背景を付けたものです。

プログラムの記述自体はなんだか暗号めいていますが，これがFORTHという言語と思ってください。図3にサンプルプログラムの実行画面を示しておきましょう。

最後に感想です

mFORTHを使用した感想は，うまく表現できませんが，結構本気で作られたFORTH言語の処理系だということです。ICEのためのシンボルファイルを生成するユーティリティが付属していたのは驚きで，最終的にはプログラムをROMに焼かなければメーカーに対して悪いんじゃないかとさえ思ってしまいます。個人的にはFORTHインタプリタが付属していないのが寂しい気がしますが，これが19,800円とは超破格値もいいところです。このmFORTHは従来のコンピュータ言語とはちょっと毛色の違うFORTHに触れて，人生観を変えてみたいという人にはお勧めかもしれません。

サイバースティックで遊ぶ

不思議な環境ソフトの世界

Kuwano Masahiko
桒野 雅彦

発売以来，売れ行き好調のサイバースティック。でも，これをアフターバーナー専用のオモチャにしておくのはあまりにももったいない話。というわけで，使って楽しむサイバースティック活用講座をお届けすることにしましょう。

7月号ですでに紹介されていますが，噂のアナログジョイスティック「サイバースティック」(CZ-8NJ:23,800円)が6月に発売になりました。売れ行きはかなり好調なようで，1カ月ほど前，ふらりと立ち寄った秋葉原の電気街でも，うず高く積まれた箱には「予約済み」のステッカーがペタペタと貼られて並んでいました。

マウスからアナログスティックへ

そのサイバースティック。これまでの，メーカー純正の機器としてはもちろん，一般的に売られている周辺機器としてもあまり見かけないものでした。現在，パソコンのオプション入力装置として最も普及しているジョイスティックやジョイパッドのたぐいは，いずれも4方向にスイッチが付いていて，スティックが傾くとその方向のスイッチがONになるようにし，これをジョイスティックポートから直接読み取る方法をとっています。斜めの方向は縦と横のスイッチが同時にONになっているかどうかを見て判断するわけです。

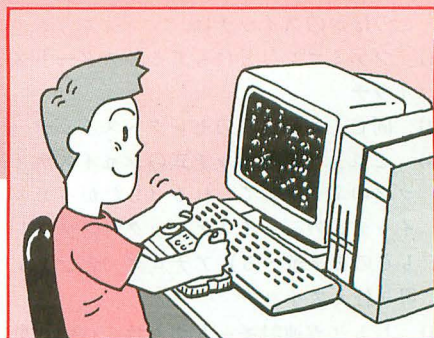
これに対し，サイバースティックは読み取りの精度をずっと細かくし，前後左右の傾きをそれぞれ128段階のデータとして読み出せるようにしています。つまり，スティックを少し傾けたのか，大きく傾けたのかという細かい情報までを読み出せるようにしているわけです。画面などで1024ドットなどという言葉を聞いていると，128段階(左端から右端まで動かして256段階）というのはキメが荒いような気がするかもしれませんが，操作する立場からすればスティックの傾けられる範囲を256分割するということですから，デジタル的なガクガクした感じはまったくありません。

真上から見るとこれだけスイッチが並んでいる

実際にサイバースティックのデータ読み取り用のBASICの外部関数を自分で作って，読み取ったデータを表示してみると，ほんの少しボディをつついたり本体を持ち上げたりしただけでデータが変化するのがわかりました。こうなると確かにアナログジョイスティックと呼べる細かさです。

考えてもみればスティックも，スロットルも，ハンドルの回転角も，およそ操作と名のつくものはみなアナログ情報です。ゲームとは，抽象化（単純化）した仮想空間の現実への投影作業と考えれば，より高度なものになるにつれてコントロールもアナログ的に行うようになるのは自然な流れなのです。

業務用のビデオゲームでも初期のブロック崩しで可変抵抗器が使われて以来，アナログ方式で動かすものはごく一部に限られており，大部分はスペースインベーダー以来，伝統的に2または4方向のスイッチを使うようになっていました。それが最近のいわゆる体感もののような高級機では，人間が行うあらゆる操作をアナログ的に取り込んで反応するようになってきています。

現在のパソコンの多くは標準入力装置としてキーボードしか持っていないということもあってか，接続されるジョイスティックもやはり4方向スイッチ方式が主流のようです。確かにこれまでの実績が示すように，このスイッチだけでもかなりのことができます。しかしゲームの種別によってはスイッチの入った方向に一定速度で移動するだけではどうしても不満が残るものが出てきます。大きく素早く動かしたり，少し位置を修正したりといった，細かなコントロールを必要とするアルカノイドなどでは操作性の面からジョイスティックに見切りをつけ，次第にマウスを選ぶようになってきました。

そうして発売開始以来3年を経て，X68000の実力がいかんなく発揮されるにつれ，ゲーム用としてだけでなく，手軽なアナログデータ入力装置としても，そろそろアナログ方式のジョイスティックが欲しいなあと，漠然とはこの私も思っていたのですが，あまり誰も見向きもしてくれそうにありませんでした。しかし，そこに登場したのが業界の初物軍団。なにがなんでも業界初！世界初！ 楽しくなければパーソナルではない！ とまでいったかどうかは知りませんが，とにかくサイバースティックはシャープの手によって世に出たわけなのです。

まずは外見のご紹介

最初にちょっと断っておきますが，サイバースティックは左右のスティックとも根元からはずれ，入れ替えることもできますが，ここでは買ってきた状態，すなわち右が4方向に動くスティック，左が前後に動くスロットルという配置を前提にお話します。

サイバースティックを見ると，まず最初に大きなスティックとスロットルが目につき，その間に押しボタンの大きいのが3個，小さいのが2個，スライドスイッチが4個，スライドボリュームが2つ，しかも右のスティックには押しボタンが2個，スライド式のスイッチがひとつ，左のスティック(スロットル）には押しボタンがひとつと，トグル式のスイッチがひとつと，とにかく一面スイッチの山です。

この山を見るなり，あのー，私は腕が2本，手の指が各5本しかないんですけどさてどうしましょ，となってしまいそうですが，これらのうちコンピュータ側への入力データとして扱われるのは8個だけで，これまでお馴染みのAトリガ，Bトリガは真ん中の大きなスイッチボタンか右側のスティックのトリガスイッチのどちらも使えるようになっています。これでプレイヤー側の入力データとして受け付けてくれるもの

は6つになります。

このほかのスイッチは，

1) ファミコンなどにもある，スタートスイッチ
2) 同じくお馴染みのセレクトスイッチ
3) これまでのスイッチ式のジョイスティックのふりをしてくれる（したがってスイッチ式のジョイスティック対応ソフトもそのまま遊べる）アナログ／デジタル切り替えスイッチ
4) Bトリガ連射モード切り替え（連射/単発/自動連射）スイッチ
5) Aトリガ連射ON/OFFスイッチ
6) A/Bボタン入れ替えスイッチ

となっています。

2つのスライドボリュームはそれぞれAボタンとBボタンの連射スピードコントロールに使われています。

アナログデータ処理

サイバースティックとX68000の接続には，これまでのジョイスティックと同様，ジョイスティックポートをそのまま使っています。ジョイスティックポートの信号線を巧妙に使って，大量のデータを分割してやり取りするようにしています。このやり取りをハードウェアだけで行うこともできなくはありませんが，そんな苦労をするよりはワンチップマイコンを載せてしまったほうが小型，簡単，安価，多機能，フレキシブル，というわけで，サイバースティックにはCPUがしっかり鎮座しています。

スイッチだけで6ビット分，それに加えてアナログチャンネルがあるのですから，サイバースティックから送られるデータ量は従来のスイッチ（デジタル）式のジョイスティックに比べてたいへん多くなっています。このデータの受け渡しのためにちょっとした手順を踏む必要があるのですが，これは後述することにして，まずは送られてくるデータを見てみることにしましょう。

まずはスイッチ類ですが，これはすべてひとつが1ビットのデータになります。離すと1で，押すと0になります。サイバースティックには6つのトリガボタンがありますが，実は送られてくるデータとしては10ビットが用意されていて，あと4つまでボタンを追加することができるようになっています。これは隠し機能というよりは，今回のスティックでは付ける場所がなかったためのようです。

続いて楽しみなアナログデータです。スティックの傾きはそれぞれ（スティックの縦方向／横方向の傾き，スロットルの傾き）8ビットのデータで送られてきます。左右方向は左端が0，右端がFFH，前後方向は奥のほうが0で手前側がFFHになっています。スティックから手を離すとセンターに戻りますが，このときのデータは80H，あるいは7FHちょうどというわけにはいかず，どうしても若干の誤差が出てきます。背面に調整用の半固定抵抗もありますが，アナログ部が結構敏感にできているのと，スティックのグリップ自体だけでかなりのモーメントがあるので，パネルをコツコツと叩いただけでも1や2はデータが変わります（振動センサや加速度計にも使えるかもしれない）。この誤差はソフトで吸収する必要があります。

アナログデータは今回のサイバースティックでは3つですが，こちらも実はもうひとつ隠されているチャンネルがあり，これもデータとして送られてきます（いつでも0だけど）。スロットル側の左右方向データでしょう。あちこち表に出てこないものがたくさんあって妙な感じですが，スティックをはずして，コネクタのピン数を数えるとなるほどねと思わせるものがあります。アナログ2つとスイッチが6個，グランドがひとつでちょうど9ピンだから，スイッチのひとつは連射ON/OFF制御用だからデータとして送られるのは5個か……，右と左でちょうど10個になったりするのです。

データの渡し方

サイバースティックから送ってくるデータは10ビットのトリガスイッチと，8ビット×4チャンネルのアナログデータの計42ビットもあります。一方，ジョイスティックポートのほうはというと9本のピンのうち，2本は電源ですから信号ラインとして使えるのは7本しかありません。この7本を使って，42ビットのデータの受け渡しを行うことになります。サイバースティックでは，3本の信号線でタイミングをとって，4ビットデータを11回送る方法を使っています。

普通のジョイスティックでは，スティックの方向を読むために使われている4本のラインをデータ用とし，トリガーボタン読み出し用の2本をサイバースティックからのステータス（ACK信号，H/L信号），残る1本をX68000からサイバースティックへのデータ転送開始要求信号（REQ信号）にしています。このようにすると，信号の向きが通常のスイッチ式のジョイスティックのときと同じになりますので，多少怪し気なインタフェイスを持ったパソコンとつないだときにデジタルモードとアナログモードを間違えたとしても単に動作しないだけで，なんらかのダメージが与えられたり，壊れたりする恐れはまったくなく，安心して使うことができます。

REQ信号はX68000からの転送開始要求としてだけでなく，信号を元に戻すタイミングによって転送タイミングが4段階に切り替わるようになっています。遅くしても特にメリットがあるわけではありませんし，X68000の場合には最高速でも問題なく取り込めます。この機能はX68000用ではなく，プロセッサの能力が低かったり，ウェイト数が多かったりして，ついてこられない機種とつなぐためでしょう。

サイバースティックは，一度REQ信号を受け取るとそのあとの11回のデータ送出は相手のタイミングを見ることなく，一方的に垂れ流してきます。したがって，機器の能力によってはついてこられない場合も考えられるわけです。

ACK（通信制御）信号は，サイバースティックがデータを揃えたことを示すもので，X68000はこの信号が0（Lレベル）のときにデータを読み込みます。

H/L信号は，サイバースティックが出しているデータが偶数番目のデータであるか，奇数番目のデータであるかを示すものです。

アナログジョイスティック誕生前史

単純にジョイスティックの傾きを読み取るだけの，アナログジョイスティックというものがこれまでなかったわけではありません。今日のパソコンの元祖ともいえる，AppleIIのジョイスティックインタフェイスはアナログでしたから，スティックの根元にボリュームを付け，トリガボタンを付けたものがずいぶん前から売られていました。私もMZ-700にIC2個を使ってジョイスティックポートに可変抵抗器（ボリューム）をつないだこともありました。分解能を問わないのであれば，ジョイスティックポートからアナログデータを読むのはさほど難しいことではありません。

ただ，そのころのジョイスティックは，いかにもボリュームに棒切れを付けましたというだけのチャチなしろもので，スティックを傾けたら，手を離しても自動的にセンターに戻るような機能さえなく，サイバースティックのような本格的なものとは，ほど遠い存在でした。どちらかというと，マウスが登場する以前の，簡易ポインティングデバイスという感じでしか捕らえられていなかったようです。

ハードウェアで転送を行うのであれば，ACKのほかにH/L信号を使う必要はありません（ACKの立ち下がりでラッチすればよいわけですから）が，ソフトでポートを読んでいる場合には，ACKだけですと非常に難しくなります。データを確実に受け取るためにはデータを読んだあと，ACKが1に戻り，再び0に戻るのを確認する必要があります。ところが，ACKが1になっている時間は4μs（最速モード時）しかありません。ちょっともたついていると1になったのを確認できないままになり，データをとりこぼすことになります。H/LはACKが1の期間に切り替わるようなので，1回目は H/L＝ACK＝0，2回目はH/L＝1，ACK＝0 という条件を見るだけで，簡単に取り込むことができるわけです。

7月号の「C調言語講座 PRO-68K」に紹介されたプログラムでは，H/Lが変わったのを確認したあとにACK＝0を確認していますが，波形をシンクロスコープで確認したところ，このように分けて見なくてはならないのは偶数番目のデータのときで（偶数番目ではACKとH/Lが同時に動く），奇数番目のときは両方を同時に見て，両方とも0という条件で判断しても構わないようです。

データの受け渡しの手順は次のようになります。

1）REQ＝0とする
2）H/L＝ACK＝0になるのを待ち，データを引き取る
3）REQ＝1にする（最速モードになる）
4）H/L＝1，ACK＝0になるのを待ち，データを引き取る
5）H/L＝0，ACK＝0になるのを待ち，データを引き取る

以下，4）と5）を4回繰り返し，合計11回の転送を終了します。その後，4ビットごとに分解されたデータを編集し直して扱いやすいデータにすればよいわけです。

デジタル式の場合のように，ポートを見ればいつでもデータが入っているというわけにはいかないので，面倒な気がするかもしれませんが，伝送プロトコルと呼べるようなシャレたことをしているわけでもなく，また関数やデバイスドライバにしてしまえば，ずっと使い回しがききますから，2回目以降はなにも考えずに便利に使うことができます。

外部関数とサンプルプログラム

7月号で，アナログジョイスティックのサポートソフトとしてデバイスドライバが「C調言語講座PRO-68K」で発表されましたが，ずいぶん気合が入っていたので，いきなりあれを見せられると，サイバースティックを使うのはずいぶんと難しいことのように思えてしまうでしょう。私も「うっ！」となってしまったひとりです。そこで，もう少し気楽に使えそうなX-BASICの外部関数を作ってみました（リスト3）。

使い方は至って簡単。まずプログラムの先頭でastset( )を実行しておきます。読み出しはキャラクタ型の配列を用意して，

char c(5)
astick(c)

のようにすればそれで終わりです。このとき，配列の各要素には図1のようなデータが入っています。

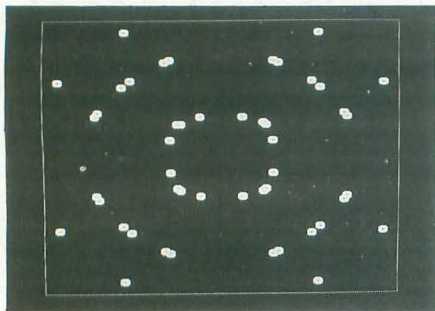
最後に，astrst( )を実行しておきます。BASICの外部関数に限定するのであれば，astset( )，astrst( )は，初期化時や systemを実行するときに自動的に呼ばれるようにすればよいのですが，コンパイルする人もいるだろうということで，別の関数にしておきました。

この関数を使って，簡単なサンプルプログラムを作ってみました。ひとつは，ジョイスティックの傾きをRGB，HSVの色情報に割り当てて，スティックとスロットルの動きに合わせた色で四角形を表示するものです（リスト1）。もうひとつは編集部で遊んでいるうちにできてしまった，スプライトが画面いっぱいにスティックに反応して踊り回るプログラム（リスト2）です。放置しておくと勝手に跳ね回り，スティックを動かすと影響を受けて動きが大きく変わります。基本のスプライト6個の動きを基に48個のスプライトが上下，左右，斜めなどに対称に動く姿は，踊っているという感覚がぴったりです。まっ，一種の環境ソフトみたいなものだと思ってください。

まず，スプライトの進行方向と，速度の

図1

リスト1

```
1000 /*
1010 /* サイバースティックチェックプログラム
1020 /* スティックの位置で色を変えます
1030 /* 右がRGB、左がHSVによるものです
1040 /*
1050 char c(5)
1060 int i
1070 astset()
1080 screen 1,3,1,1
1090 locate 22,20:print"H              R"
1100 locate 22,21:print"S              G"
1110 locate 22,22:print"V              B"
1120 repeat
1130     astick(c)
1140     fill(145,200,245,300,hsv(c(0)*100/133,c(1)/8,c(2)/8))
1150     fill(255,200,355,300,rgb(c(0)/8,c(1)/8,c(2)/8))
1160     locate 22,10
1170     for i=0 to 5
1180         print right$("0"+hex$(c(i)),2);" ";
1190     next
1200     locate 24,20:print c(0)*100/133:locate 38,20:print c(0)/8
1210     locate 24,21:print c(1)/8:locate 38,21:print c(1)/8
1220     locate 24,22:print c(2)/8:locate 38,22:print c(2)/8
1230 until (c(4) and 2)=0
1240 astrst()
```

リスト2実行画面

初期値を6個，スティックで決めます。スティックを傾けながらBボタンを押せば確定です。すべての向きが決まるとスプライトが動き始めます。壁にぶつかると跳ね返りますが，このとき少しだけ速度が落ち，次第に動きが遅くなります。ちょっと細工があるので，ときどき乱数で速度が与えられ，いつまでも動き続けます。ここでAボタンを押しながらスティックを動かすと，動きが大きく変わるようになっています。小さいプログラムですから，この動きに関しては言葉で説明するよりも，実際にプログラムを打ち込んでもらったほうがわかりやすいでしょう。

とにかく，見ているものを不思議な気持ちにしてくれるソフトに仕上がってしまいました。これにゆったりとした音楽を用意すれば，最近はやりのストレス解消のためのα波がどうしたこうしたという，立派な(?)環境ソフトとしても使えると思います。最初は，ただX68000のスプライトを使って，サイバースティックでいくつかのキャラクタをピコピコ動かそうとしただけなんですけどね。

*　　　　*

マイコンをパソコンと呼び変え，それまでの楽しければなんでもありといった自由奔放な機械がオフィスオートメーションの掛け声とともに，ただのビジネスツールにされてしまってずいぶん日時が経過したように思います。某社のとったこの戦略は，それなりの成功を収め，大きなシェアを握るに至りました。しかし，これがために6年もたっても相変わらずCPUとメモリ容量の変更ばかりという事態を招いたということは，パーソナルコンピュータの進歩という面からすれば大きな損失でした。おかげで現在のほとんどのパーソナルコンピュータが6年以上も前の設計そのままの，MS-DOS付きワープロ＋αになってしまっている有り様です。

そのような状況のなかで生まれた大胆なパーソナルコンピュータに，これまた大胆な発想のもとに生まれた周辺機器が，このサイバースティックです。フルに使えば8ビット×4チャンネルのアナログデータと，10ビットのデジタルデータを渡せるこの方式なら，3次元データ入力もできる（そういえば，AppleIIにはバーサライターなんていうものがあったっけ)。ハンドルを付けてもいい。ロボットのコントローラにもいいかも。そうそう，サラウンドしちゃうなんてどう？　と，使い方はいくらでも考えられそうです。あー，貯金なんてあまりないというのに……，でもきっとこのOh!Xが発売されるころには，買ってしまっているんだろうなあ。

リスト2

```
1000 int x(5),y(5),dx(5),dy(5),g(5)
1010 char c(5)
1020 int count
1030 int i,sdx,sdy
1040 astset()
1050 screen 1,3,1,1
1060 console 0,31,0
1070 sp_init()
1080 sprite_pattern()
1090 sp_disp(1):sp_on()
1100 sdx=0:sdy=0
1110 for i=0 to 5
1120 box(2,2,506,506,1989)
1130     repeat
1140         astick(c)
1150         line(256,256,256+sdx/2,256+sdy/2,0)
1160         sdx=(c(1)-&H80)*4:sdy=(c(0)-&H80)*4
1170         line(256,256,256+sdx/2,256+sdy/2,65535)
1180         locate 5,i:print "#";i;":";"dx=";sdx,"dy=";sdy
1190     until (c(4) and &H40)=0
1200     line(256,256,256+sdx/2,256+sdy/2,0)
1210     dx(i)=sdx:dy(i)=sdy
1220     x(i)=25600:y(i)=25600:g(i)=5
1230     repeat
1240         astick(c)
1250     until (c(4) and &H40)
1260 next
1270 cls
1280 box(2,2,506,506,1989)
1290 repeat
1300  for i=0 to 5
1310    astick(c)
1320    if (c(4) and &H80)=0 then dx(i)=dx(i)+c(1)-&H80:dy(i)=dy(i)+c(0)-&H80
1330    sp_move(i,x(i)/100,y(i)/100)
1340    sp_move(i+6,492-x(i)/100,y(i)/100)
1350    sp_move(i+12,492-x(i)/100,492-y(i)/100)
1360    sp_move(i+18,x(i)/100,492-y(i)/100)
1370    sp_move(i+24,y(i)/100,x(i)/100)
1380    sp_move(i+30,492-y(i)/100,x(i)/100)
1390    sp_move(i+36,492-y(i)/100,492-x(i)/100)
1400    sp_move(i+42,y(i)/100,492-x(i)/100)
1410    if x(i)+dx(i)>49000 then {
1420        x(i)=98000-x(i)-dx(i):dx(i)=-dx(i)*9/10
1430    } else {
1440        if x(i)+dx(i)<100 then x(i)=200-x(i)-dx(i):dx(i)=-dx(i)*9/10 else x
(i)=x(i)+dx(i)
1450    }
1460    if y(i)+dy(i)>49000 then {
1470        y(i)=98000-y(i)-dy(i):dy(i)=-dy(i)*9/10
1480    } else {
1490        if y(i)+dy(i)<100 then y(i)=200-y(i)-dy(i):dy(i)=-dy(i)*9/10 else y
(i)=y(i)+dy(i)
1500    }
1510    dy(i)=dy(i)+g(i)
1520    if y(i)>48500 and x(i)<26000 and x(i)>25000 then dx(i)=-rnd()*300+150:d
y(i)=-rnd()*1400-200
1530   next
1540 until (c(4) and 2)=0
1550 end
1560 func sprite_pattern()
1570   dim char c(255)
1580   int i
1590   c={
1600     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1610     0,0,0,0,15,15,15,15,15,15,15,15,0,0,0,0,
1620     0,0,0,15,4,4,4,4,4,4,4,4,15,0,0,0,
1630     0,0,15,4,7,7,7,7,7,7,7,7,4,15,0,0,
1640     0,15,4,7,7,6,6,6,6,6,6,7,7,4,15,0,
1650     0,15,4,7,6,6,7,6,6,7,6,6,7,4,15,0,
1660     0,15,4,7,6,7,12,13,13,12,7,6,7,4,15,0,
1670     0,15,4,7,6,6,13,14,14,13,6,6,7,4,15,0,
1680     0,15,4,7,6,6,13,14,14,13,6,6,7,4,15,0,
1690     0,15,4,7,6,7,12,13,13,12,7,6,7,4,15,0,
1700     0,15,4,7,6,6,7,6,6,7,6,6,7,4,15,0,
1710     0,15,4,7,7,6,6,6,6,6,6,7,7,4,15,0,
1720     0,0,15,4,7,7,7,7,7,7,7,7,4,15,0,0,
1730     0,0,0,15,4,4,4,4,4,4,4,4,15,0,0,0,
1740     0,0,0,0,15,15,15,15,15,15,15,15,0,0,0,0,
1750     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1760   }
1770   for i=0 to 255
1780   sp_def(0,c)
1790   next
1800 endfunc
```

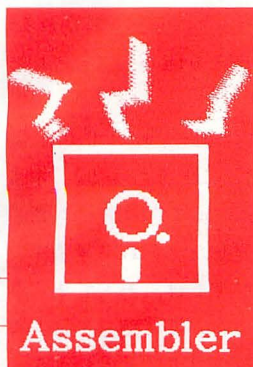
リスト3

```
0000  D0 00 00 00 03 86 41 53  : ED
0008  54 49 43 4B 00 00 C0 01  : EC
0010  00 00 02 C2 74 65 78 74  : 89
0018  00 4B C0 02 00 00 00 00  : 0D
0020  64 61 74 61 00 4B C0 03  : A8
0028  00 00 00 00 62 73 73 00  : 48
0030  C0 04 00 00 00 00 73 74  : AB
0038  61 63 6B 00 B2 01 00 00  : E2
0040  00 F4 5F 61 73 74 69 63  : 67
0048  6B 00 B2 01 00 00 00 8C  : AA
0050  5F 61 73 74 73 65 74 00  : F3
0058  B2 01 00 00 00 C0 5F 61  : 33
0060  73 74 72 73 74 00 20 01  : 61
0068  00 00 00 00 42 01 00 00  : 43
0070  00 40 42 01 00 00 00 40  : C3
0078  42 01 00 00 00 40 42 01  : C6
-----------------------------------
SUM:  DA 67 1C BA 27 84 BD D1  7651

0080  00 00 00 40 42 01 00 00  : 83
0088  00 40 42 01 00 00 00 40  : C3
0090  42 01 00 00 00 40 42 01  : C6
0098  00 00 00 40 42 01 00 00  : 83
00A0  00 42 42 01 00 00 00 58  : DD
00A8  42 01 00 00 00 6C 10 2B  : EA
00B0  00 00 00 00 00 00 00 00  : 00
00B8  00 00 00 00 00 00 00 00  : 00
00C0  00 00 00 00 4E 75 61 73  : 97
00C8  74 69 63 6B 00 61 73 74  : F3
00D0  73 65 74 00 61 73 74 72  : 06
00D8  73 74 00 00 42 01 00 00  : 2A
00E0  00 64 42 01 00 00 00 68  : 0F
00E8  42 01 00 00 00 6A 10 07  : C4
00F0  00 34 FF FF FF FF FF FF  : 2E
00F8  42 01 00 00 00 7C 42 01  : 02
-----------------------------------
SUM:  62 60 9C ED 74 DD EB 8C  54C5

0100  00 00 00 8C 42 01 00 00  : CF
0108  00 C0 30 00 00 00 00 04  : F4
0110  10 17 22 6F 00 0C 43 E9  : F0
0118  00 0A 2F 09 61 6C 58 8F  : F6
0120  4E 75 42 A7 FF 20 58 8F  : B2
0128  23 C0 46 01 00 00 00 78  : A2
0130  10 0B 20 7C 00 E9 A0 07  : 47
0138  10 BC 00 09 20 39 46 01  : 75
0140  00 00 00 78 10 0D 6B 08  : 08
0148  2F 01 00 00 FF 20 58 8F  : 36
0150  70 00 41 F9 46 01 00 00  : F1
0158  02 BC 10 0D 31 40 00 02  : 4E

0160  4E 75 42 A7 FF 20 58 8F  : B2
0168  23 C0 46 01 00 00 00 78  : A2
0170  10 0B 20 7C 00 E9 A0 07  : 47
0178  10 BC 00 08 20 39 46 01  : 74
-----------------------------------
SUM:  D3 96 22 DB 67 6B DA 33  E2AF

0180  00 00 00 78 10 0D 6B 08  : 08
0188  2F 01 00 00 FF 20 58 8F  : 36
0190  70 00 41 F9 46 01 00 00  : F1
0198  02 BC 10 1F 31 40 00 02  : 60
01A0  4E 75 61 22 0C 80 00 00  : D2
01A8  00 00 66 0E 20 2F 00 04  : C7
01B0  2F 00 61 00 00 C4 58 8F  : 3B
01B8  70 00 41 F9 46 01 00 00  : F1
01C0  02 BC 10 0D 31 40 00 02  : 4E
01C8  4E 75 42 A7 FF 20 58 8F  : B2
01D0  23 C0 46 01 00 00 00 78  : A2
01D8  10 07 40 E7 00 7C 07 00  : C1
01E0  41 F9 46 01 00 00 02 AA  : 2D
01E8  10 73 22 7C 00 E9 A0 01  : AB
01F0  24 7C 00 E9 A0 07 34 3C  : A0
01F8  00 04 70 00 36 3C 03 E8  : D1
-----------------------------------
SUM:  86 16 6A BB FE EA 53 04  7ECA

0200  14 BC 00 08 10 11 12 00  : 0B
0208  02 00 00 60 57 CB FF F6  : 79
0210  67 02 60 5C 10 C1 14 BC  : C6
0218  00 09 36 3C 00 64 08 11  : F8
0220  00 05 56 CB FF FA 66 02  : 87
0228  60 50 12 11 08 01 00 06  : E2
0230  57 CB FF EC 67 02 60 42  : 18
0238  10 C1 36 3C 00 64 10 11  : C8
0240  12 00 02 00 00 60 57 CB  : 96
0248  FF F6 67 02 60 2C 10 C1  : BB
0250  51 CA FF C8 76 00 46 DF  : 7D
0258  14 BC 00 09 22 39 46 01  : 7B
0260  00 00 00 78 10 11 6B 08  : 0C
0268  2F 01 00 00 FF 20 58 8F  : 36
0270  20 03 4E 75 76 01 43 F9  : 99
0278  46 01 00 00 02 4E 10 05  : AC
-----------------------------------
SUM:  4F 29 E9 C4 64 A7 0C 1F  4732

0280  60 DC 76 02 43 F9 46 01  : 37
0288  00 00 02 7C 10 07 60 D2  : C7
0290  20 6F 00 04 43 F9 46 01  : 16
0298  00 00 02 AA 10 D5 12 29  : CC
02A0  00 02 02 01 00 0F E9 41  : 3E
02A8  14 29 00 06 02 02 00 0F  : 56

02B0  82 02 10 C1 12 29 00 03  : 93
02B8  02 01 00 0F E9 41 14 29  : 79
02C0  00 07 02 02 00 0F 82 02  : 9E
02C8  10 C1 12 29 00 04 02 01  : 13
02D0  00 0F E9 41 14 29 00 08  : 7E
02D8  02 02 00 0F 82 02 10 C1  : 68
02E0  12 29 00 05 02 01 00 0F  : 52
02E8  E9 41 14 29 00 09 02 02  : 74
02F0  00 0F 82 02 10 C1 12 29  : 9F
02F8  00 00 02 01 00 0F E9 41  : 3C
-----------------------------------
SUM:  25 CB 21 AF 4B 61 8C C0  BB5F

0300  14 29 00 01 02 02 00 0F  : 51
0308  82 02 10 C1 12 29 00 0A  : 9A
0310  02 01 00 0F 10 C1 4E 75  : A6
0318  83 54 83 43 83 6F 81 5B  : 6B
0320  81 45 83 58 83 65 83 42  : 4E
0328  83 62 83 4E 82 AA 82 A2  : 06
0330  82 C8 82 A2 82 DD 82 BD  : 0C
0338  82 A2 82 BE 82 AF 82 C7  : DE
0340  81 45 81 45 00 00 83 54  : 63
0348  83 43 83 6F 81 5B 81 45  : 5A
0350  83 58 83 65 83 42 83 62  : 6D
0358  83 4E 82 AA 82 C6 82 DC  : A3
0360  82 C1 82 BF 82 E1 82 C1  : 2A
0368  82 BD 81 45 81 45 81 45  : 91
0370  81 45 00 00 30 00 00 00  : F6
0378  00 11 10 06 00 00 00 00  : 27
-----------------------------------
SUM:  B2 93 B9 E7 69 7F E4 2E  DCD8

0380  00 00 00 0F 00 00 00 00  : 0F
0388  00 00 00 00 00 00 00 00  : 00
0390  00 00 00 00 00 00 00 00  : 00
0398  00 00 00 00 00 00 00 00  : 00
03A0  00 00 00 00 00 00 00 00  : 00
03A8  00 00 00 00 00 00 00 00  : 00
03B0  00 00 00 00 00 00 00 00  : 00
03B8  00 00 00 00 00 00 00 00  : 00
03C0  00 00 00 00 00 00 00 00  : 00
03C8  00 00 00 00 00 00 00 00  : 00
03D0  00 00 00 00 00 00 00 00  : 00
03D8  00 00 00 00 00 00 00 00  : 00
03E0  00 00 00 00 00 00 00 00  : 00
03E8  00 00 00 00 00 00 00 00  : 00
03F0  00 00 00 00 00 00 00 00  : 00
03F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  00 00 00 0F 00 00 00 00  BD6C
```

リスト4

（このプログラムをアセンブルするにはCコンパイラの付属ファイルが必要です）

```
================== astick.s ==================
  1: *------------------------------------------------------------
  2: *- アナログ・ジョイスティック（サイバー・スティック）
  3: *- 読み込み関数
  4: *-
  5: *- 1989-06-17 M.Kuwano, No rights reserved
  6: *............................................................
  7: *- 呼び出し方（例）：
  8: *- char c(5)
  9: *- astick(5)
 10: *- 入るデータは・・
 11: *-          c(0).....右側スティック左右
 12: *-          c(1).....右側スティック前後
 13: *-          c(2).....左側スティック前後（スロットル）
 14: *-          c(3).....予備（将来用？）
 15: *-          c(4).....トリガ・ボタン
 16: *-          c(5).....トリガ・ボタン
 17: *- と、なります
 18: *- 配列は6バイト以上確保してください。
 19: *- 配列のサイズチェックをやっていないので、あらぬところまで
 20: *- 書き込んでしまいます。
 21: *............................................................
 22: *- いつもながらインタプリタ／コンパイラ共用です
 23: *-
 24: *------------------------------------------------------------
 25:            .include        doscall.mac
 26:            .include        fdef.h
 27:            .globl          _astick
 28:            .globl          _astset
 29:            .globl          _astrst
 30:
 31: IOCS              equu             $0f
 32:
 33: PPI_PORT_A equ              $e9a001
 34: PPI_PORT_B equ              $e9a003
 35: PPI_PORT_C equ              $e9a005
 36: PPI_CWR           equ              $e9a007
 37:
 38: RQ_ASSERT  equ              $8
 39: RQ_NEGATE  equ              $9
 40:
 41: TIME_LIMIT1       equ              1000
 42: TIME_LIMIT2       equ              100
 43:            .text
 44:            .even
 45: *
 46: *  インフォメーション・テーブル
 47: *
 48: *
 49:            .dc.l           AS_INIT
 50:            .dc.l           AS_RUN
 51:            .dc.l           AS_END
 52:            .dc.l           AS_SYS
 53:            .dc.l           AS_BRK
 54:            .dc.l           AS_CTRL_D
 55:            .dc.l           AS_RES1
 56:            .dc.l           AS_RES2
 57:            .dc.l           PTR_TOKEN
 58:            .dc.l           PTR_PARAM
 59:            .dc.l           PTR_EXEC
 60:            .dc.l           0,0,0,0,0
 61:
 62: AS_RES1:
 63: AS_RES2:
 64: AS_END:
 65: AS_BRK:
 66: AS_CTRL_D:
 67: AS_INIT:
 68: AS_RUN:
 69: AS_SYS:
 70:            rts
 71:
 72: *
 73: *  トークン・テーブル
 74: *
 75: PTR_TOKEN:
 76:            .dc.b           'astick',0
 77:            .dc.b           'astset',0
 78:            .dc.b           'astrst',0
 79:            .dc.b           0
 80:            .even
 81: *
 82: *  パラメータ・テーブル
 83: *
 84: PTR_PARAM:
 85:            .dc.l           ASTICK_PAR
 86:            .dc.l           ASTSET_PAR
 87:            .dc.l           ASTRST_PAR
 88:
 89: *
 90: *  パラメータIDテーブル
 91: *
 92: ASTICK_PAR:
 93:            .dc.w           ary1_c
 94:            .dc.w           void_ret
 95: ASTSET_PAR:
```

```
 96:            .dc.w           void_ret
 97: ASTRST_PAR:
 98:            .dc.w           void_ret
 99:
100: *
101: *  関数アドレステーブル
102: *
103: PTR_EXEC:
104:            .dc.l           astick
105:            .dc.l           _astset
106:            .dc.l           _astrst
107:
108: *
109: *  スタック・バッファ
110: *
111: SPBUF:
112:            .ds.l           1
113:
114:            .even
115: *
116: *  アナログ・ジョイスティック読みだし(インタプリタ用)
117: *
118: astick:
119:            movea.l         12(sp),a1
120:            lea             10(a1),a1
121:            move.l          a1,-(sp)
122:            bsr             _astick
123:            addq.l          #4,sp
124:            rts
125:
126: *
127: *  アナログ・ジョイスティック用に、PC4を1にする
128: *
129: _astset:
130:            clr.l           -(sp)                   *    SPBUF = _SUPER(0);
131:            dc.w            _SUPER
132:            addq.l          #4,sp
133:            move.l          d0,SPBUF
134:            movea.l         #PPI_CWR,a0             *    *ppi_cwr = RQ_NEGATE;
135:            move.b          #RQ_NEGATE,(a0)
136:            move.l          SPBUF,d0                *    _SUPER(SPBUF);
137:            bmi             astset_already_super
138:            move.l          d1,-(sp)
139:            dc.l            _SUPER
140:            addq.l          #4,sp
141:    astset_already_super:
142:            moveq.l         #0,d0
143:            lea.l           AS_RETVAL,a0
144:            move.w          d0,2(a0)
145:            rts
146:
147: *
148: *  プログラム終了後、PC4を0に戻しておかないと
149: *  デジタル・モード用のソフトのうち動かなくなるも
150: *  のがでてくるらしい
151: *
152: _astrst:
153:            clr.l           -(sp)                   *    SPBUF = _SUPER(0);
154:            dc.w            _SUPER
155:            addq.l          #4,sp
156:            move.l          d0,SPBUF
157:            movea.l         #PPI_CWR,a0             *    *ppi_cwr = RQ_ASSERT;
158:            move.b          #RQ_ASSERT,(a0)
159:            move.l          SPBUF,d0                *    _SUPER(SPBUF);
160:            bmi             astrst_already_super
161:            move.l          d1,-(sp)
162:            dc.l            _SUPER
163:            addq.l          #4,sp
164:    astrst_already_super:
165:            moveq.l         #0,d0
166:            lea.l           AS_RETVAL,a0
167:            move.w          d0,2(a0)
168:            rts
169:
170: *
171: *  アナログ・ジョイスティック読みだし(コンパイル時用)
172: *
173: _astick:
174:            bsr             get_astick              * get_astick();
175:            cmpi.l          #0,d0
176:            bne             _astick_end
177:            move.l          4(sp),d0
178:            move.l          d0,-(sp)
179:            bsr             aj_compile
180:            addq.l          #4,sp
181:            moveq.l         #0,d0
182:    _astick_end:
183:            lea.l           AS_RETVAL,a0
184:            move.w          d0,2(a0)
185:            rts
186:
187: *
188: *  アナログ・ジョイスティックデータ取り込み
189: *
190: * a0 Buffer_pointer
191: * a1 PPI_PORT_A
192: * a2 PPI_CWR
193: * d0 data
194: * d1 data
195: * d2 Loop counter
196: * d3 Timeout counter
197: *
198: get_astick:                                             * get_astick() {
199:            clr.l           -(sp)                   *    SPBUF = _SUPER(0);
200:            dc.w            _SUPER
201:            addq.l          #4,sp
202:            move.l          d0,SPBUF
203:            move.w          sr,-(sp)                *    PUSH(SR);
204:            ori.w           #$0700,sr               *    disable_trap();
205:            lea.l           AS_TMP_BUF,a0           *    buffer_pointer = AS_TMP_BUF;
206:            movea.l         #PPI_PORT_A,a1          *    ppi_port_a     = PPI_PORT_A;
207:            movea.l         #PPI_CWR,a2             *    ppi_cwr        = PPI_CWR;
208:
209:            move.w          #4,d2                   *    loop_counter = 4;
210:            moveq.l         #0,d0                   *    joydata = 0;
211:            move.w          #TIME_LIMIT1,d3         *    timer = TIME_LIMIT1
212:            move.b          #RQ_ASSERT,(a2)         *    *ppi_cwr = RQ_ASSERT
213:    _astick_1:
214:            move.b          (a1),d0                 *    while (((data = *ppi_port_a & 0x60)
 != 0) && timer--)
215:            move.b          d0,d1                   *               ;
216:            andi.b          #$60,d0
217:            dbeq            d3,_astick_1
218:            beq             _astick_ok_1            *    if (!timer)
219:            bra             _astick_not_ready       *       goto(_astick_not_ready);
220:    _astick_ok_1:
221:            move.b          d1,(a0)+                *    *buffer_pointer++ = data;
222:            move.b          #RQ_NEGATE,(a2)         *    *ppi_cwr = RQ_NEGATE;
223:
224:    _astick_2:                                      *    do {
225:            move.w          #TIME_LIMIT2,d3         *       timer = TIME_LIMIT2;
226:    _astick_21:
227:            btst.b          #5,(a1)                 *       while(!(*ppi_port_a & 0x20) && t
imer--)
228:            dbne            d3,_astick_21           *             ;
229:            bne             _astick_22              *       if (!timer)     goto _astick_tim
eout;
230:            bra             _astick_timeout
231:    _astick_22:                                     *       while(((data = *ppi_port_a) & 0x
40) && timer--)
232:            move.b          (a1),d1
233:            btst            #6,d1                   *             ;
234:            dbeq            d3,_astick_21
235:            beq             _astick_ok_2            *       if (!timer)     goto _astick_tim
eout;
236:            bra             _astick_timeout
237:    _astick_ok_2:
238:            move.b          d1,(a0)+                *       *buffer_pointer++ = data;
239:            move.w          #TIME_LIMIT2,d3         *       timer = TIME_LIMIT2;
240:    _astick_3:
241:            move.b          (a1),d0                 *       while(((data = (*ppi_port_a & 0x
60)) != 0) && timer--)
242:            move.b          d0,d1                   *             ;
243:            andi.b          #$60,d0
244:            dbeq            d3,_astick_3            *       if (!timer)
245:            beq             _astick_ok_3            *             goto (_astick_timeout);
246:            bra             _astick_timeout
247:
248:    _astick_ok_3:
249:            move.b          d1,(a0)+                *       *buffer_pointer++ = data;
250:            dbra            d2,_astick_2            *    } while (loop_counter--);
251:
252:            moveq.l         #0,d3                   *    retstat = 0;
253:    _astick_exit:
254:            move.w          (sp)+,sr                *    sr = POP(); /* 割り込みフラグを戻す
  */
255:            move.b          #RQ_NEGATE,(a2)         *    *ppi_cwr = RQ_NEGATE;
256:            move.l          SPBUF,d1                *    _SUPER(SPBUF);
257:            bmi             _astick_already_super
258:            move.l          d1,-(sp)
259:            dc.l            _SUPER
260:            addq.l          #4,sp
261:    _astick_already_super:
262:            move.l          d3,d0
263:            rts                                     * }
264:
265:    _astick_not_ready:
266:            moveq.l         #1,d3                   *    data = 1;
267:            lea.l           AS_MSG_NOT_READY,a1     *    msgptr = AS_MSG_NOT_READY;
268:            bra             _astick_exit
269:
270:    _astick_timeout:
271:            moveq.l         #2,d3                   *    data = 2;
272:            lea.l           AS_MSG_TMOUT,a1         *    msgptr = AS_MSG_TMOUT;
273:            bra             _astick_exit
274: *
275: *  アナログ・ジョイスティックデータ編集
276: *
277:
278: aj_compile:                                             * aj_compile(pack_data) {
279:            movea.l         4(sp),a0
280:            lea.l           AS_TMP_BUF,a1
281:
282:            move.b          2(a1),d1        *    pack_data[0] = joy_raw_data[6] | (joy_raw_d
ata[2] << 4);
283:            andi.b          #$f,d1
284:            asl             #4,d1
285:            move.b          6(a1),d2
286:            andi.b          #$f,d2
287:            or.b            d2,d1
288:            move.b          d1,(a0)+
289:
290:            move.b          3(a1),d1        *    pack_data[1] = joy_raw_data[7] | (joy_raw_d
ata[3] << 4);
291:            andi.b          #$f,d1
292:            asl             #4,d1
293:            move.b          7(a1),d2
294:            andi.b          #$f,d2
295:            or.b            d2,d1
296:            move.b          d1,(a0)+
297:
298:            move.b          4(a1),d1        *    pack_data[2] = joy_raw_data[8] | (joy_raw_d
ata[4] << 4);
299:            andi.b          #$f,d1
300:            asl             #4,d1
301:            move.b          8(a1),d2
302:            andi.b          #$f,d2
303:            or.b            d2,d1
304:            move.b          d1,(a0)+
305:
306:            move.b          5(a1),d1        *    pack_data[3] = joy_raw_data[9] | (joy_raw_d
ata[5] << 4);
307:            andi.b          #$f,d1
308:            asl             #4,d1
309:            move.b          9(a1),d2
310:            andi.b          #$f,d2
311:            or.b            d2,d1
312:            move.b          d1,(a0)+
313:
314:            move.b          0(a1),d1        *    pack_data[4] = joy_raw_data[1] | (joy_raw_d
ata[0] << 4);
315:            andi.b          #$f,d1
316:            asl             #4,d1
317:            move.b          1(a1),d2
318:            andi.b          #$f,d2
319:            or.b            d2,d1
320:            move.b          d1,(a0)+
321:
322:            move.b          10(a1),d1       *    pack_data[5] = joy_raw_data[10];
323:            andi.b          #$f,d1
324:            move.b          d1,(a0)+
325:
326:            rts                                     * }
327:
328:
329: AS_MSG_NOT_READY:
330:            .dc.b           'サイバー・スティックがいないみたいだけど・・',0
331:            .even
332: AS_MSG_TMOUT:
333:            .dc.b           'サイバー・スティックがとまっちゃった・・・・',0
334:            .even
335: AS_TMP_BUF:
336:            .ds.b           12
337: AS_ACK_BUF:
338:            .ds.b           5
339:            .even
340: AS_RETVAL:
341:            .dc.w           0
342:            .dc.w           0
343:            .dc.w           0
344:            .end
```

正しいフィルタの作り方(前編)

Murata Toshiyuki 村田 敏幸

さて，あるデータを「フィルタ」にかける，というのはよくやることですが，今回は前後編にわたり，半角の英小文字を大文字に変換するフィルタを作っていきます。ワンステップずつ理解しながら，当連載初の実用プログラムをぜひ自分のものにしてください。

今回の題材は“フィルタ”である。今月と来月の2回にわたって，半角の英小文字を大文字に変換する（ほかの文字は素通りさせる）フィルタを例に「フィルタはこう作る！」というところを見てもらおうと思う。プログラムは仮にUPPER.Xと名づけよう。これは僕たちが作る初の実用プログラム（使い道があるかどうかは別にして）になる。

とはいえ最初から即使えるプログラムを作ろうなんて欲ばるとあとで後悔することになるから，まずはできる範囲で作る。そして一応動くものができたら動作試験をしてみて，不備を探し出しては修正していく。この試す・直すというサイクルは，プログラムを作りデバッグする過程そのものであり，今回はそのあたりもちょっと強調してみるつもりでいる。なお，フィルタに関しては109ページのコラム参照。

小文字→大文字変換の手順

標準入出力を使った文字の取扱いはもう飽きるほどやった。あとは英小文字を大文字に変換する方法を押さえれば，とりあえず“小文字→大文字変換フィルタ”の原型を作ることができる。小文字→大文字変換の手法は先月最後のプログラムで示してあるが，ここでもう一度詳しく話しておこう。

この処理は次の2つの段階からなる。

1) 対象となる文字が英小文字かどうか調べる
2) そうであれば大文字に変換する

しごく当然。

第1のステップである英小文字かどうかの判断はASCIIコードの比較で行う。“a”〜“z”の文字には61_H〜$7A_H$の連続したASCIIコードが割り当てられているから，下限の61_Hと上限の$7A_H$との都合2回比較し，その大小関係から小文字とそうでないものに選り分けることができる。つまり，次のような手順になる。

1-1) 下限である61_Hと比較する。61_H未満であれば小文字ではない
1-2) 上限である$7A_H$と比較する。$7A_H$より大きければ小文字ではない
1-3) 1-1)と1-2)のチェックに引っかからなければ小文字である。

第2のステップもまたASCIIコードレベルでの単純な操作だ。“A”〜“Z”の文字には41_H〜$5A_H$の ASCIIコードが割り当てられており，これは小文字のコードとちょうど20_Hずれている。ということは，小文字のASCIIコードから20_Hを引けば大文字に変換できるわけだ。

以上の考えをそのままサブルーチンの形にするとリスト1のようになる。このサブルーチン toupperはd0.bにASCIIコードを入れて呼び出すと，小文字→大文字の変換を行って，結果をd0.bに入れて戻る。

4〜7行が小文字かどうかを調べている部分だ。この判定の結果，英小文字と判断されたら8行で20_Hを引いて大文字に変換する。なお，ここで使っているbcs, bccに関しては“無符号数の大小比較”を参照してもらいたい。

小文字→大文字変換の処理は20_Hを引く代わりに，DF_HとANDを取っても同じ結果が得られる。理由は各自考えよう。

試作する：UPPER.X第1版

材料が揃ったところで，UPPER.Xの第1版をリスト2に示す。プログラム自体は簡単なものだからとくに解説すべき点もない。DOSコールgetcで1文字読み込み，英小文字だったらサブルーチン toupp

リスト1 英小文字→大文字変換サブルーチン

```
1: *半角英小文字→英大文字変換サブルーチン
2:
3: toupper:
4:          cmpi.b  #'a',d0          *英小文字か?
5:          bcs     toupr0           *
6:          cmpi.b  #'z'+1,d0        *
7:          bcc     toupr0           *
8:          subi.b  #$20,d0          *小文字なら大文字に変換
9: toupr0:  rts                      *サブルーチンからリターン
```

リスト2　UPPER.S（その1）

```
 1: *        英小文字→英大文字変換フィルタ   第1版
 2:
 3:          .include       doscall.mac
 4: *
 5:          .text
 6:          .even
 7: *
 8: ent:
 9:          lea.l   mysp,sp           *spの初期化
10:
11: loop:    DOS      _GETC            *1文字入力
12:          bsr     toupper           *小文字→大文字変換
13:          move.w  d0,-(sp)          *1文字出力
14:          DOS      _PUTCHAR         *
15:          addq.l  #2,sp             *
16:          bra     loop              *えんえんと繰り返す
17: *
18: *英小文字→英大文字変換サブルーチン
19: toupper:
20:          cmpi.b  #'a',d0           *英小文字か?
21:          bcs     toupr0            *
22:          cmpi.b  #'z'+1,d0         *
23:          bcc     toupr0            *
24:          subi.b  #$20,d0           *小文字なら大文字に変換
25: toupr0: rts                        *サブルーチンからリターン
26: *
27:          .stack
28:          .even
29: *
30: mystack:
31:          .ds.l   256               *スタック領域
32: mysp:
33:          .end
```

erで大文字に変換してからputcharで標準出力に書き出す。これを無限ループの中で延々と繰り返している。

では，アセンブル後，

A>UPPER

で起動し，適当にキーボードから文字を打ち込んで，正しく動作するかどうか確認してもらいたい。英小文字が大文字に変換されるかどうか確認するのはもちろん，“そうでない文字”が化けたりしないかどうかなどの点についても忘れずにチェックしてほしい。いい加減に作ってあるだけあって，早々に問題点が見つかるだろう。

まず，リターンキーを押したときの改行動作がおかしい。カーソルは行頭に戻るだけで，次の行には移動してくれない。そのため，続けて入力した文字が重ねて表示されてしまい，大変に見苦しい。この原因を探るために，プログラムの動作を追ってみよう。

リターンキーを押すと$0D_H$というコードが入力される。これは英小文字ではないので，大文字への変換処理は行われずに素通りし，そのまま標準出力に吐き出される。$0D_H$は“カーソルを行頭に移動する”コントロールコードだから，これによりカーソルは行の先頭に戻る。こののち，プログラムはループして再び文字の入力待ちになり，次に入力された文字は重ねて表示されることになる。

Human68kでは改行は$0D_H$と$0A_H$という2つのコードで行うことになっているから，$0A_H$の“カーソルを次の行に移動する”動作が欠けている分，改行が中途半端になってしまっているわけだ。

結局，この症状はOSの改行コード（$0D_H$＋$0A_H$）とキーボードから入力される改行コード（$0D_H$）との差に原因があった。本来，このような場合はOSがつじつまを合わせてくれる（キーボードから$0D_H$を入力したら自動的に$0D_H$＋$0A_H$の入力に変換してくれる）べきだと思うのだが，少なくともDOSコールgetcにはそういうサービスはないようだ。

そこで，$0D_H$が入力されたら$0D_H$に続けて$0A_H$も“入力されたことにしてやる”ことを考える。“本来ならOSにやってほしかった”ことをプログラム側で肩代わりしてやろうという発想だ。けれど無条件に$0A_H$をつけるような単純な処理では，あとでまた別の問題が生じることになる。この件に関してはもう少し動作試験をしてから結論を出すことにしよう。

さて，UPPER.Xの不備その2。こんなことは動作試験をするまでもないのだが，無限ループになっている関係でいつまでたってもプログラムが終了しない。これも立派なバグといえるだろう。DOSコールgetc内ではブレイクチェックが効くから，BREAKキーを押せば止まることは止まるのだが，スマートとはいえないし，標準入力をファイルにリダイレクトしたときにどうなるかが心配だ。あとでテストしてみる必要がある。

動作試験の心構え

上で見つけた2つの動作不良は，ちょっと試してみれば誰にでも見つけられる類いのものだった。しかし，現実のプログラムのバグはこんな素直なものばかりとは限らない。もう大丈夫だと思ったプログラムに，ある日突然ひょんなことから虫が見つかるというのはよくある話である。あとで困りたくなければ，考え得る限りの状況を想定して事前に厳しく動作試験するほかない。見つからないバグは潰しようがないのだ。そのためには動作試験のときにどれだけの“状況”を考えつけるかが鍵になる。

ここで一時脱線して，一般的な動作試験時のコツというかポイントを簡単にまとめておこう。これはなにもマシン語に限った話ではない。

たとえば，高級言語で平方根を求めるプログラムなり関数なりを作ったとする。平方根ぐらい関数として用意されているだろうが，あえてなにやら得体のしれないアルゴリズムを駆使して作ったものとしよう。プログラムは一応動くようになり，動作試験をすることになった。読者ならどこをどうやってテストするだろうか。

2とか3とか平方根がいくつになるか暗記している数をパラメータとして与え,「ひとよひとよにひとみごろ」とか「ひとなみにおごれや」とかブツブツいいながら，結果が正しいかどうか見比べるぐらいのことは誰でもやりそうだ。もう少し多くの数につ

いて結果を求め，数表なり電卓なり，すでに動作が確認されているほかのプログラム（BASICの関数など）と比較すればなお安心だろう。このとき可能であれば，目で見て比べるのではなく，プログラムでチェックできればより確実になる。

次に“誤った入力があった場合”に何が起こるかも確かめておかなければならない。これはプログラムにとっては“入力が適切なときに正しい結果を返す”ことと同じくらい重要なことだ。プログラムにエラーチェックを組み込んであるのなら，故意にエラーを引き起こすようなデータを与えてみて，エラーチェック自体が正しく機能するかどうかを確認しておく。平方根プログラムの場合であれば，パラメータが負の数だったら（複素数を扱うことを考えない限り）エラーとしてはじくように作られているはずだから，－1かなんかを食わせてみて“予定どおりエラーになる”ことを確かめる。エラーチェックを組み込むことを忘れていた場合には，この段階でプログラムの異常が発見されるだろう。

が，やっかいなことに，エラーチェックを忘れたときにはそもそも“プログラムに不当な入力があるかもしれない”ことも忘れてしまっている場合が多い。プログラムを作ったときの先入観を動作試験に持ち込んでしまうと，肝心なことを見落としてしまうものだ。動作試験に臨むときには，すぱっと頭を切り換える必要がある（口でいうほど簡単なことではないが）。この場合の必殺技は“制作者とは完全に別な頭”つまり他人に動作試験してもらうことだろう。人数は多ければ多いほどよい。「完成したら君にも使わせてあげるからさー」とかなんとかいって友だちをまるめ込もう。

ひとり黙々とデバッグしているのであれば，無理をしてでもさらに“状況”をひねり出してみる。

“極端な場合”というのも要チェック事項である。この平方根プログラムも，もしかすると非常に大きな数を与えると演算過程でオーバーフローして，誤った答えを返すかもしれないし，絶対値の非常に小さな数を与えると，予想外に精度が低下するという症状が見つかるかもしれない。これらは実際に大きな数や小さな数を与えて動作試験しなければ，潜在的なバグとしてあとあとまで残ってしまうことになる。

架空の平方根プログラムでは関係ないが，“条件判断の境界付近”も重点チェック項目のひとつだ。“10以下”と判断すべきところを“10より大きい”にしてしまうようなミスは単純なだけに見つけにくいが，実際に10の前後の数をデータとして与えて実験してみれば一目瞭然だ。

それから，ループが正しい回数だけ回るかどうかも確かめておかなければならない。BASICのforループのような一定回数回すループで，ループカウンタの初期値が1少なかったり多かったりするのはかなり（マシン語では非常に）危険なバグである。また，不定回数のループ（X-BASICのwhile～endwhileやdo～whileで表されるような構文）では，“最初から条件が成立している場合は1度もループを通らない

フィルタ

狭義では，標準入力からデータを読み込み，適当な処理を加えて結果を標準出力に書き出すプログラムのことを“フィルタコマンド”ないしは単に“フィルタ”という。この場合，扱うデータはテキスト（文書，文字データ）であることが仮定される。

標準入出力は通常コンソール（キーボードとCRT）に割りつけられているから，その状態でのフィルタは“キーボードからデータを読み込み（処理を加えて）結果を画面に表示する”だけのプログラムとして機能する。が，DOSのサービス（正確にはCOMMAND.Xの機能）であるリダイレクションを利用して，

```
A>FILTER <FILE1
```

という具合に標準入力をファイルにリダイレクトしてやれば，“FILE1からデータを読み込んで結果を画面に出す”ようになるし，

```
A>FILTER >FILE2
```

と標準出力をリダイレクトすれば“キーボードから読み込んだデータを処理して結果をFILE2に書き出す”プログラムに早変わりする。

さらには，

```
A>FILTER <FILE1 >FILE2
```

と入出力ともにリダイレクトすれば，“FILE1からデータを読み込んで結果をFILE2に書き出す”こともできるし[1]，

```
A>FILTER <FILE1 >PRN
```

なら，“FILE1からデータを読み込み，結果をプリンタに出力する”ことになる。

フィルタ関係でリダイレクションと並んでもうひとつ重要なDOSのサービス（これも本当はCOMMAND.Xの機能）が“パイプ”で，これは複数のフィルタを連結するものだ。たとえば，

```
A>FILTER1 <FILE1 | FILTER2
```

とやれば，FILTER1の出力がそのままFILTER2の入力となる[2]。もちろん，フィルタはさらに3つ4つといくらでもつなげて使うこともできる[3]。

1つひとつのフィルタは単純な機能しか持っていなくても，このようにリダイレクションを活用したりパイプでつないだりすることで，より複雑な処理を行うことができるようになる。逆にいえば，個々のフィルタは単純な処理だけを行うように作ればすむし，むしろ単純であればあるだけ応用が利くともいえる。

1）フィルタをリダイレクションと組み合わせて使う場合，

```
A>FILTER <FILE1 >FILE1
```

のように入力ファイルと出力ファイルに同一のものを指定すると期待どおりの処理が行われないから気をつけよう。

2）Human68kでのパイプは一時ファイルを介して行われる。つまり，1段目のフィルタの出力はこっそりファイルに書き出され，そのファイルが次のフィルタの入力となる。この一時的に作成されたファイルはコマンドの実行後自動的に削除される。要するに，

```
A>FILTER1 | FILTER2
```

は，内部では

```
A>FILTER1 >TMP
A>FILTER2 <TMP
A>DEL TMP
```

と同等な処理に置き換えて実行されている。なお，この一時ファイルが生成されるパスは，

```
A>TEMP A:¥
```

のように指定しておくことができる。この指定がない場合にはカレントディレクトリに一時ファイルが生成される。RAMディスクが使えるのであれば，一時ファイルをRAMディスク上に生成するようにしておくと速くて静かである。

3）実際にはCOMMAND.Xのコマンド行の最大入力文字数＝255文字によって制限を受ける。

のか，それとも1度は回るのか”という点がチェック項目になる。これは設計時に考慮すべきことだが，実際に何通りかの条件下でプログラムを動かしてみればはっきりするだろう。

プログラムを作り慣れないころは，どこからテストすればよいのか，なかなかわからないかもしれない。が，いくつもプログラムを作っていくうちに，“こないだはこういうことがあったから今度はこうかな”という勘(?)が働くようになる。これを推し進めていけば，動作試験時ではなく設計の段階でバグを潰せるようになる寸法だ。

では，ここで話したことを教訓にUPPER.Xをいま一度厳しく動作試験してみてもらいたい。致命的なバグ，“一部の文字が不当に化ける”という症状が見つかるはずだ（見つからないようではまだまだ頭が固い）。

このUPPER.X 3つ目のバグ“全角文字を入力すると，変な文字に化けて出力される場合がある”というものだ。たとえば“敏幸”と入力してみると“媛幸”のように化ける。これは漢字コードの関係で起こる症状であり，詳しくはコラム“日本語の呪い”で説明しておく。

しつこく動作試験してみる

さて，キーボードからの入力ですらこれだけ問題があったわけだが，フィルタであるからにはリダイレクションを交えて実験してみる必要がある。次のようにして，標準入力を自分自身のソースファイルにリダイレクトして実行してみよう。キーボードからデータを入力した場合とはまた違った症状がいくつか見られると思う。

```
A>UPPER <UPPER.S
```

まず，改行は正しく行われている。正しく動作しているのはよいことだが，キーボードからデータ入力した場合と結果が違うのはやはりおかしい。また懸念であった“無限ループになっていることの影響”は，しっかり“ファイルの最後まで処理した時点でそのまま黙り込んでしまう”という症状になって現れる。しかも黙り込む直前に画面がクリアされてし

読者がHuman68kのver 2.0を使用しているのであれば，BREAKキーによりこの沈黙状態から抜けることができる。が，Human68kのver1.0を使用している場合はなぜかBREAKキーが効かないので，INTERRUPTスイッチを押して強制終了してもらいたい。これはver1.0のマイナーなバグだと思う。

無符号数の大小比較

条件分岐の基本中の基本である，“比較してみた結果が，等しければ分岐する”とか“等しくなければ分岐する”手法は何度も顔を出し，cmp後に，beqやbne を使えばよいのだった。ここでbeqとbneの実体は“Zフラグがセットされていれば（1ならば）分岐する”命令と，“Zフラグがリセットされていれば（0ならば）分岐する”命令だということ，68000にはZ以外にCやNなどのフラグもあったことを思い出してもらうと，Z以外のフラグを条件とする分岐命令もあるだろうということが想像できる。

bcs，bccはCフラグを条件とする分岐命令で，csとccはそれぞれCarry Set，Carry Clear の略だ。言葉どおり，C（キャリ）フラグがセットされているとき分岐する命令と，リセットされているとき分岐する命令である。

どういう用途に使うかは，Cフラグがどんな場合に変化するか考えてみるとわかると思う。Cフラグは加算の結果が演算サイズを越えて繰り上がりが発生した場合や，減算の結果，桁借りが発生した場合に立つ（＝セットされる＝1になる）フラグだから，まず，加減算のオーバーフローの判定に使えるだろう。

たとえばd1.wがFFFF$_H$でd0.wが 0002$_H$のときに，

```
        add.w     d1, d0
```

を実行すると，加算結果は10001$_H$になってしまい，オペレーションサイズ(この場合はワード)で表現できる範囲の数を越えるので，Cフラグがセットされる（もちろんd0.wには下位ワードの0001$_H$だけが残る）。ワードからの繰り上がりが生じたとき分岐したければ，この直後に

```
        bcs       ～
```

を置けばよいし，繰り上がりが生じなかったとき分岐したければ，

```
        bcc       ～
```

を使えばよい。

また同様の設定で，

```
        sub.w     d1, d0
```

を実行すると，d0.wには0003$_H$という結果が得られる。これを無符号演算と考えると，2から65535(＝FFFF$_H$）が引けないので，上位から桁借りし，10002$_H$－FFFF$_H$と見なして演算したものと見なせる。そして，この桁借りが発生したことを表すために，やはりCフラグがセットされ(意味的にはCarryではなくBorrow)，直後に，

```
        bcs       ～
```

を置けば，“桁借りが生じたときに分岐する”ことができる。

次にCフラグが比較時に持つ意味について考えてみる。比較命令cmpは“減算を行い，フラグだけを変化させて結果を捨てる”命令だから，subとフラグの変化はまったく同じだ。subでは減算の結果，上位桁からの桁借りがあったときにCフラグがセットされたわけだが，桁借りが生じたということは被減数が減数より小さかったということだ。すると，

```
        cmp.w     d1, d0
```

実行後“(無符号数で考えて）d0.wがd1.wより小さい”ときにはCフラグが立つ。つまり，

```
        cmp.w     d1, d0
        bcs       ～
```

は，d0.wのほうがd1.wより小さいときに分岐する，

```
        cmp.w     d1, d0
        bcc       ～
```

なら，d0.wがd1.w以上のとき分岐する，という意味になる。bcs，bccの導入により，無符号数の大小比較が行えるようになったわけだ。

68000にはさらにCフラグとZフラグを組み合わせて条件分岐する賢い命令が用意されている。これまでに出てきたものと併せて，下にまとめておく。

bhi	C＝0かつZ＝0のとき分岐
bcc	C＝0のとき分岐
beq	Z＝1のとき分岐
bne	Z＝0のとき分岐
bls	C＝1またはZ＝1のとき分岐
bcs	C＝1のとき分岐

ここでなにかを感じ取ってもらえれば話は早いのだが，ひらめくものがなければ次を見てほしい。これは，

```
        cmp       X, Y
```

実行後のフラグ変化とXとYの大小関係との関連を示したものだ。

Y>X	C＝0，Z＝0
Y≧X	C＝0，Z＝0またはC＝0，Z＝1
Y＝X	C＝0，Z＝1
Y≠X	C＝0，Z＝0またはC＝1，Z＝0
Y≦X	C＝1，Z＝0またはC＝0，Z＝1
Y<X	C＝1，Z＝0

すべての可能性を網羅したのでちょっとごちゃごちゃしているが，下線部が必要十分条件である。

つまり，上に挙げた6つの条件分岐命令により，任意の(無符号数の)大小比較が行える。ちなみにbhiのhiはHIgh，blsのlsはLess or Sameの意味だ。

さて，しつこく“無符号数の場合”ということを強調してきた。ということは，符号付き数の比較にはここで紹介した方法は使えないということだ。では，符号付き数の比較を行う方法はないのかというと，実はちゃんとある。が，符号付き数を扱う場面はあまりないので，いまここで覚える必要はぜんぜんない（よって紹介もしない)。まとめて覚えようとすると，かえって混乱してしまうことにもなりかねないが，ゆとりのある人は各自調べてみてほしい。

まうという不可解なオマケももれなくついてくる。

ここまでの実験でUPPER.Xの現バージョンにおける問題点がかなりはっきりしてきた。

1) BREAKキーを押さないと止まらない。キーボードからデータを入力する場合はともかく，標準入力をリダイレクトした場合はファイルの最後まで処理したら停止してくれないと困る

2) キーボードから入力する場合に改行動作が正しくない（標準入力をファイルにリダイレクトすると正しく改行する）

3) 入力をファイルに切り換えた場合，ファイルの最後でなぜか画面がクリアされてしまう

4) 全角文字の入力が考慮されていない

次の節ではこれらの問題点を一気にクリアする。

改良する:UPPER.X第2版

まず，ファイルの最後まで処理したら自動的に終了するようにしよう。そのためには，ファイルの最後を検出する方法を知らなくてはならない。

Human68kのテキストファイルは一般に次のような構造をしている。

1) 各行は$0D_H$，$0A_H$で終わる

2) ファイルの最後は$1A_H$という1バイトのコードで表す

試しにUPPER.SをDUMP.Xで覗いてみると（図1）このとおりの構造だということが確認できるだろう。

この新しい情報によって，プログラムをいつ終了させたらよいかがわかる。つまり，$1A_H$が入力されたら，実行を終えるようにすればよい。これはgetcにより入力された1バイトデータを$1A_H$と比較して処理を振り分けるような数行の追加で実現できるだろう。この変更により，データをキーボードから入力するときも，CTRL+Z を押せば終了するようになる。

ここで，ついでに$1A_H$というコードについてもう少し調べてみよう。『Human68k ユーザーズマニュアル』のコントロールコード一覧を見てみると，$1A_H$は画面クリアのコントロールコードになっている。さっきの実験でファイルの最後まで処理した時点で画面が消去されたのは，このファイルエンドコード兼画面消去コードの$1A_H$がフィルタを素通りし画面に出力されてしまったためだったのだ。入力が$1A_H$だったら即座にプログラムを終了する（$1A_H$は出力しない）ようにすれば，謎の画面クリアの問題も片づくことになる。

次に改行動作の不備を直す。入力の$0D_H$を特別扱いするわけだが，単純に$0A_H$を付け加えるのではう

図1

```
00000000  2A 09 89 70 8F AC 95 B6 8E 9A 81 A8 89 70 91 E5 *.英小文字→英大
00000010  95 B6 8E 9A 95 CF 8A B7 83 74 83 42 83 8B 83 5E 文字変換フィルタ
00000020  81 40 91 E6 82 50 94 C5 0D 0A 0D 0A 09 2E 69 6E 　第1版......in
00000030  63 6C 75 64 65 09 64 6F 73 63 61 6C 6C 2E 6D 61 clude.doscall.ma
                                   :
                                   :
00000210  0A 6D 79 73 74 61 63 6B 3A 0D 0A 09 2E 64 73 2E .mystack:....ds.
00000220  6C 09 32 35 36 09 09 2A 83 58 83 5E 83 62 83 4E l.256..*スタック
00000230  97 CC 88 E6 0D 0A 6D 79 73 70 3A 0D 0A 09 2E 65 領域..mysp:....e
00000240  6E 64 0D 0A 1A                                  nd...
```

リスト3　UPPER.S（その2）

```
 1: *           英小文字→英大文字変換フィルタ　第2版
 2:
 3:             .include        doscall.mac
 4: *
 5:             .text
 6:             .even
 7: *
 8: ent:
 9:             lea.l   mysp,sp         *spの初期化
10:
11: loop:       DOS     _GETC           *1文字入力
12:
13:             cmpi.b  #$1a,d0         *ファイルエンドコードか？
14:             beq     done            *そうなら終了
15:
16:             cmpi.b  #$0a,d0         *LFコードか？
17:             beq     loop            *そうなら無視
18:
19:             cmpi.b  #$0d,d0         *CRコードか？
20:             beq     cr_lf           *そうならLF,CRにして出力
21:
22:             cmpi.b  #$80,d0         *80Hより小さければ
23:             bcs     hankaku         *       ASCIIコード
24:             cmpi.b  #$a0,d0         *80H以上A0H未満なら
25:             bcs     zenkaku         *       シフトJISの1バイト目
26:             cmpi.b  #$e0,d0         *A0H以上E0H未満なら
27:             bcs     hankaku         *       ASCIIカタカナ
28:                                     *E0H以上なら
29:                                     *       シフトJISの1バイト目
30: *
31: zenkaku:
32:             move.w  d0,-(sp)        *シフトJISの1バイト目を
33:             DOS     _PUTCHAR        *       そのまま出力
34:             addq.l  #2,sp           *
35:
36:             DOS     _GETC           *もう1バイト持ってくる
37:             move.w  d0,-(sp)        *シフトJISの2バイト目も
38:             DOS     _PUTCHAR        *       そのまま出力
39:             addq.l  #2,sp           *
40:
41:             bra     loop            *繰り返す
42: *
43: hankaku:
44:             bsr     toupper         *小文字→大文字変換
45:
46:             move.w  d0,-(sp)        *1文字出力
47:             DOS     _PUTCHAR        *
48:             addq.l  #2,sp           *
49:
50:             bra     loop            *繰り返す
51: *
52: cr_lf:
53:             move.w  d0,-(sp)        *d0にはCRコードが入っている
54:             DOS     _PUTCHAR        *CRコードを出力
55:             move.w  #$0a,(sp)       *LFコードを出力
56:             DOS     _PUTCHAR        *
57:             addq.l  #2,sp           *
58:
59:             bra     loop            *繰り返す
60: *
61: done:
62:             DOS     _EXIT           *終了
63: *
64: *英小文字→英大文字変換サブルーチン
65: toupper:
66:             cmpi.b  #'a',d0         *英小文字か？
67:             bcs     toupr0          *
68:             cmpi.b  #'z'+1,d0       *
69:             bcc     toupr0          *
70:             subi.b  #$20,d0         *小文字なら大文字に変換
71: toupr0:     rts                     *サブルーチンからリターン
72: *
73:             .stack
74:             .even
75: *
76: mystack:
77:             .ds.l   256             *スタック領域
78: mysp:
79:             .end
```

まくない。というのは，入力をファイルに切り換えているときには，各行の終わりの$0D_H$，$0A_H$が，$0D_H$，$0A_H$，$0A_H$ に変換されてしまうことになるからだ。

入力がキーボードかファイルかを調べることができれば処理を振り分けることができるし，実際そう

することも可能なのだが，ここではもっと簡単な方法を採ることにした。

1)　0DHが入力されたら，0DH，0AHの2バイトに変換して出力する

2)　0AHが入力されたら，捨てる（出力しない）

これにより，0DHが単独で現れた場合にも，0DH，0AHが連続して現れた場合にも正しく対応できるようになる。ファイルの途中にぽつんと0AHが入っていたりする可能性がないとはいえないから，これは100%の方法ではないが，実用上の問題はないという判断だ。

最後に全角文字を考慮して英小文字→大文字変換周りを練り直す。いままでは文字を単純に1バイトずつ処理していたために，2バイトを組にして扱うべきシフトJISコードが不当に化けてしまっていたわけだから，入力された文字がシフトJISコードかどうかを調べ，そうであれば素通りさせるような処理を付け加えればよいだろう。手順は次のようになる。

1)　1バイト持ってくる

2)　それがシフトJISコードの1バイト目でなければ，英小文字→大文字変換してから結果を出力し，1)へ戻る

3)　シフトJISコードの1バイト目であれば，無条件に素通りさせる（変換は一切行わない）。続いてもう1バイト持ってきて，これも素通りさせる。それから1)へ戻る

で，以上の発見された限りの不備を直したUPPER.Xの修正版がリスト3だ。13～14行がファイルの終端かどうかを調べる部分，16行あたりが改行周りの処理になっている。また，22～27行が全角文字の1バイト目かどうかのチェックで，ここでやっていることは，

1)　80Hより小さいコードは半角文字

2)　80H～9FHならばシフトJISコードの1バイト目

3)　A0H～DFHならば半角（カタカナ）

4)　E0H以上ならシフトJISコードの1バイト目

という一連の比較と処理の振り分けだ。本来シフトJISコードの1バイト目は81H～9FHまたはE0H～EFHの範囲だから2)と4)のチェックは正しくないように見えるが，X68000では8000H台とF000H以降の2バイトコードで各種の半角文字を表すことになっているのでこれでよいのだ。

プログラムの各部の働きがわかったら，またさっきのようにあれこれと実験してみよう。入力がキーボードのときのリターンキーの動作は正しいか，標準入力をファイルにリダイレクトしたときにはファイルの最後でちゃんと止まるか，全角文字が化けたりしないか，1つひとつ確認しておくこと。

さらに改良する：UPPER.X第3版

さっきHuman68kのテキストファイルは1AHで終わるといった。しかし，実はこのファイルエンドコードはなくてもよいことになっている。試しに，

A>MORE <UPPER.S >TEST.S

を実行してもらいたい。TEST.SとUPPER.Sの内容はまったく変わらないが(TYPEしてみよう)，DIRコマンドでディレクトリをとってみると，TEST.Sのほうが1バイト短いことがわかると思う。DUMP.Xでダンプしてみれば，UPPER.Sにはあったファイルエンドコード1AHがTEST.Sにはないことも確認できるだろう。ファイル末の1AHが削られてしまうのは，MORE.Xに限らずHuman68k上のフィルタの一般的な副作用だ。

TEST.Sは，当然エディタで読み込んで編集することができるし，AS.Xでアセンブルすることもできる。このことはファイルエンドコードがついていなくてもテキストファイルとして成り立つということの証拠である。

さて，いまのところUPPER.Xはファイルエンドコードのみに頼ってファイルの終端を調べている。このため，上の話にあった“エンドコードのないテキストファイル”はきちんと処理できない。

A>UPPER <TEST.S

を実行すると，予想どおりファイルの最後まで処理した時点で黙り込んでしまう。

そこで，UPPER.Xをさらにリスト4のように修正する。初登場のDOSコールfgetcを使って標準入力からの1文字入力を行っている。詳しい話は次回にまわすが，15行からの

```
clr.w   -(sp)
DOS     _FGETC
addq.l  #2, sp
```

は，ほぼ

```
DOS     _GETC
```

と同等の処理を行うものだ。大きな違いはgetcがまったくエラーを返さないのに対して，fgetcのほうはエラーであればd0.lに負の数を入れて戻るという点だ。ファイルが終わっているのにさらにデータを読み込もうとしたときに発生するエラーを検出し，その時点でプログラムを終了しようという姑息な手である。エラーが生じたかどうかを調べているのが19行の，

```
tst.l   d0
```

であり，tstの実行によりd0.lが負であればNフラグが立つ（Nフラグが1になる）から，

```
bmi     ~
```

より処理を振り分けている。

MORE.Xは1画面ごとに表示を一時停止する拡張TYPEコマンドのようなプログラムだが，このように標準入出力ともにリダイレクトしたときには“何もしないフィルタ”として動作する。いまの場合はUPPER.SはTEST.Sにそのままコピーされる。

実はHuman68kではテキストファイルとそうでないファイル（バイナリファイル）を内部ではまったく区別していない。あくまでユーザーが勝手に区別しているに過ぎない。ところで，エンドコードがないとすると，どうやってファイルの終わりを表しているのかという疑問が生じるが，これは単純なことで，ディスクにはファイルの本体以外にファイルの長さなどの情報が別に格納してあるのだ。ほら，DIRでファイルサイズが表示されるじゃないか。

このbmiはbeqやbneの親戚で，その意味は“演算結果が負（=Nフラグが1）であれば分岐”である。ついでにいっておくと“演算結果が正（Nフラグが0）のとき分岐”させたければ，

```
bpl     ~
```

を使う。それぞれはMInusとPLusの略になっている。

厳密にはこのチェックは不十分であり，ファイルエンドに達したとき以外のエラーも同列に扱ってしまっているのはよいことではない。ま，応急手当てといったところだ。

あと，いままで数字で記述していた改行コードの類いが5～7行で定義したラベルに置き換えられている点に目を止めてもらいたい。このようにプログラム中で使う定数を意味を持った記号で書き表すようにすれば，プログラムの読みやすさが向上し，また，ミスタイプの危険も少なくすることができる。

第4版に向けて

UPPER.Xは2段階の修正を経て，だいぶまともなフィルタになってきた。しかし，まだ，完全とはいえず，バグと呼んで差し支えないような不備が残っている。

十分長いテキストファイルを用意し，

A>UPPER <FILE

を実行してみると，たぶん，気持ちの悪い症状が見られるだろう(出力結果が読めるか？)。また，リスト5に示すX-BASICプログラムを実行するとTESTというファイル名の短いファイルが生成されるから，TYPE，DUMPで内容を確認したうえで，UPPER.Xの第1～3版それぞれに対して，

A>UPPER <TEST

を実行してもらいたい。そして，この2つの実験からバグの原因を想像してみよう。

さらに，実行速度の問題がある。現バージョンのUPPER.Xはマシン語で書かれているにもかかわらず，遅いのだ。XCを持っている人は，リスト6のUPPER.Xと同等な処理を行う（ただし全角文字の入力は考慮していない）Cプログラム“C_UPPER.C”をコンパイルし，

A>UPPER <UPPER.S >TEST1

A>C_UPPER <UPPER.S >TEST2

をそれぞれ実行して両者の速度を比較してもらいたい。この結果は少々ショックである。

あと，人間工学的（？）に見て，このプログラムをもっと使いやすくできないかどうかは一度検討してみる必要がある。たとえば，フィルタはほとんど

リスト4　UPPER.S（その3）

```
 1: *        英小文字→英大文字変換フィルタ　第3版
 2:
 3:          .include        doscall.mac
 4: *
 5: CR       equ     $0d
 6: LF       equ     $0a
 7: EOF      equ     $1a
 8: *
 9:          .text
10:          .even
11: *
12: ent:
13:          lea.l   mysp,sp         *spの初期化
14:
15: loop:    clr.w   -(sp)           *1文字入力
16:          DOS     _FGETC          *
17:          addq.l  #2,sp           *
18:
19:          tst.l   d0              *エラーか？
20:          bmi     done            *そうなら終了
21:
22:          cmpi.b  #EOF,d0         *ファイルエンドコードか？
23:          beq     done            *そうなら終了
24:
25:          cmpi.b  #LF,d0          *LFコードか？
26:          beq     loop            *そうなら無視
27:
28:          cmpi.b  #CR,d0          *CRコードか？
29:          beq     cr_lf           *そうならCR,LFにして出力
30:
31:          cmpi.b  #$80,d0         *80Hより小さければ
32:          bcs     hankaku         *       ASCIIコード
33:          cmpi.b  #$a0,d0         *80H以上A0H未満なら
34:          bcs     zenkaku         *       シフトJISの1バイト目
35:          cmpi.b  #$e0,d0         *A0H以上E0H未満なら
36:          bcs     hankaku         *       ASCIIカタカナ
37:                                  *E0H以上なら
38:                                  *       シフトJISの1バイト目
39: *
40: zenkaku:
41:          move.w  d0,-(sp)        *シフトJISの1バイト目を
42:          DOS     _PUTCHAR        *       そのまま出力
43:          addq.l  #2,sp           *
44:
45:          DOS     _GETC           *もう1バイト持ってくる
46:          move.w  d0,-(sp)        *シフトJISの2バイト目も
47:          DOS     _PUTCHAR        *       そのまま出力
48:          addq.l  #2,sp           *
49:
50:          bra     loop            *繰り返す
51: *
52: hankaku:
53:          bsr     toupper         *小文字→大文字変換
54:
55:          move.w  d0,-(sp)        *1文字出力
56:          DOS     _PUTCHAR        *
57:          addq.l  #2,sp           *
58:
59:          bra     loop            *繰り返す
60: *
61: cr_lf:
62:          move.w  d0,-(sp)        *d0にはCRコードが入っている
63:          DOS     _PUTCHAR        *CRコードを出力
64:          move.w  #LF,(sp)        *LFコードを出力
65:          DOS     _PUTCHAR        *
66:          addq.l  #2,sp           *
67:
68:          bra     loop            *繰り返す
69: *
70: done:
71:          DOS     _EXIT           *終了
72: *
73: *英小文字→英大文字変換サブルーチン
74: toupper:
75:          cmpi.b  #'a',d0         *英小文字か？
76:          bcs     toupr0          *
77:          cmpi.b  #'z'+1,d0       *
78:          bcc     toupr0          *
79:          subi.b  #$20,d0         *小文字なら大文字に変換
80: toupr0:  rts                     *サブルーチンからリターン
81: *
82:          .stack
83:          .even
84: *
85: mystack:
86:          .ds.l   256             *スタック領域
87: mysp:
88:          .end
```

リスト5　テストプログラム(X-BASIC用)

```
10 /*UPPER.Xテスト用プログラム
20 int fp
30 dim char dat(9) = {
40      'a','b','c',3,'d','e','f',13,10,26
50 }
60 /*
70      fp = fopen( "test","c" )
80      fwrite( dat, 10, fp )
90      fclose( fp )
```

リスト6　C_UPPER.C(XC用)

```
 1: #include    <stdio.h>
 2: #include    <ctype.h>
 3:
 4: void main()
 5: {
 6:     int c;
 7:
 8:     while( EOF != ( c = getchar() ) )
 9:         putchar( toupper( c ) );
10: }
```

EOFはEnd Of Fileの略

の場合リダイレクトを利用して使うものだから，わざわざ，

```
A>UPPER <UPPER.S
```

と打ち込むのではなく，

```
A>UPPER UPPER.S
```

のように直接ファイル名を指定できるようにするべきではないだろうか。

●

次回は正しいフィルタの作り方（後編）として，UPPER.Xにバグ取り＋高速化＋マンマシンインタフェイスの改善を施し，また，突如出てきたDOSコールfgetcに関する補足を行う予定でいる。

もし，読者が来月まで暇をもてあますようであれば，UPPER.Xを改造して大文字を小文字に変換するフィルタLOWER.Xを作ってみたり，全角の大文字を全角の小文字に変換するフィルタを作ってみたりするのも面白いだろう。

というところで来月へと続く。

日本語の呪い

英語圏では，コンピュータが扱う文字といえば英数字と若干の記号だけだ。これはたかだか100個程度の数であり，7ビット（128通り）もあればカバーできる範囲だし，もう少し特殊な記号やセミグラフィック文字などを加えたとしても，8ビット＝1バイト（256通り）のコードで十分表現できる範囲である。対して，日本では漢字というものがあるお陰で，1バイトのコードでは足りずに，2バイトで1文字を表す漢字コードがASCIIコードと併用されている。“併用”というのが悲惨なところで，これは文字を扱うプログラムを作るうえで，ときにややこしい問題を引き起こす。本文にあるUPPER.Xの第1版は，見事にこの問題に引っかかってしまっている。常に1文字＝1バイトである英語圏のプログラマとは違って，僕たちはこの“日本語の呪い”を引きずっているのだ。

漢字コードには何種類かあるが，ふつう使われるのは“JIS漢字コード”と“シフトJIS漢字コード”だけと考えて差し支えないだろう。X68000では表向きシフトJISコードが採用されているが，内部的にはときおりJISコードを使う場面もあるようだ。

JISコードでは，第1，第2バイトともに 21_H ～ $7E_H$ の範囲の数だけが使われる。つまり 2121_H から始まり $217E_H$ の次は 2221_H に飛ぶわけだ。JISコードを考えた人は，00_H ～ $1F_H$ と $7F_H$（スタンダードなASCIIコードでは“カーソル位置の1文字削除”の意味を持つ）のコントロールコードおよび 20_H のスペースとダブらないように気を遣ったらしい。80_H 以降を使わないのは，文字を7ビットで表すようなプログラムを考慮したためだろうか。

JISコードの最大の欠点は，ASCIIコードと単純に区別できないということだ。文字列中に 41_H，42_H というバイト列があった場合，これを“AB”というASCIIコード2文字と見なしたらよいのか，JISコードの1文字“疎”と見なしたらよいのかがわからない。苦肉の策として，文字列中の漢字の前後に“漢字イン（ここから漢字ですよーという意味のコード）”と“漢字アウト（ここまでが漢字ですよーという意味のコード）”を置く方法が使われることになった。この方法では“疎疎”という文字列は

漢字イン，41_H，42_H，41_H，42_H，漢字アウト

という形で表される。

が，この方法には“データが長くなる”，“部分文字列を抜き出すのが面倒”などの欠点が残っている。

シフトJISコードはもう少しうまい方法で1バイトコード（ASCIIコード）と混在できるように作られている。シフトJISコードの1バイト目に漢字インコードの意味を併せ持たせるとでもいったらいいのだろうか。

シフトJISコードでは，第1バイトは 81_H ～ $9F_H$，$E0_H$ ～ EF_H が，第2バイトは 40_H ～ $7E_H$，80_H ～ FC_H が使われる。第1バイトには妙な隙間があるが，ここはカタカナのASCIIコード[1]が割り当てられている部分だ。つまり，シフトJISコードの1バイト目はASCIIコード 00_H ～ $7F_H$ の英数記号・コントロールコード，$A0_H$ ～ DF_H のカナと重複しないように作られている。これにより，半角英数字・カナとシフトJISコードが混在しても，しっかり区別できるというわけだ[2]。

シフトJISコードとASCIIコードが混在した文字列は，先頭から1バイトずつ順に見ていって，81_H ～ $9F_H$，$E0_H$ ～ EF_H のコードが現れたら，その次の1バイトと合わせた2バイトが漢字を表すものと判断する。JISコードを使ったときのような漢字イン，漢字アウトが必要ない点に注目してほしい。

ここで，せっかくASCIIコードと区別できるようになってはいても，プログラム側で対応していなければ意味がない。一例を挙げると，DOSコールgetcではシフトJISコードは1バイトずつ2度に分けて入力されるから，これをもとの2バイトコードに再構成するのはユーザープログラムの仕事だ。

UPPER.Xの第1版では，これを怠っていたために，たとえば“敏”（シフトJISコード 9571_H）が入力されると，

1） 1バイト目の 95_H を持ってくる。ASCIIコードの小文字の範囲外だから素通りさせる
2） 2バイト目の 71_H を持ってくる。“q”のASCIIコードだから，大文字“Q”（ASCII コード 51_H）に変換する

というように誤動作し，結果として，“敏”が“媛”（シフトJISコード 9551_H）に化けてしまう。正しくは，1バイト目を取り出した時点で，シフトJISコードだということを調べ，2バイト目も素通りさせるよう細工しなければならない。

さて，実際にはX68000ではシフトJISコードを若干拡張して使っている。これらは他機種では通用しないX68000独自の文字コードであり，あれば便利なことは確かなのだが，プログラムを作るときには問題になることもある。

まず，$EB9F_H$ ～ $EC9E_H$ は（全角）外字のコードになっている。これはシフトJISコードの空き部分だからプログラムではほかの全角文字と区別して扱う必要はない[3]。

$80XX_H$ で表されるコードは，半角1文字を表す（XXの部分にはASCIIコードが入る）。たとえば，8041_H は半角の“A”を意味する。画面に表示した場合はASCIIコード 41_H の“A”とまったく同じである。これだけなら面白くもなんともないが，8001_H ～ $801F_H$ は“コントロールコードを実行せずに，文字として表示する”のに利用できる（うまく説明しにくいのでASK の記号入力かなんかで調べてみてほしい）。また $80A0_H$ ～ $80DF_H$ にはカタカナでなくひらがなが割り当てられているので，知っていると役に立つこともあるだろう。

$F0XX_H$，$F1XX_H$ には1/4角上付き文字が，$F2XX_H$，$F3XX_H$ には同下付き文字が割り当てられている（“2”とか“A”のようなやつ）。$F0XX_H$ と $F1XX_H$，$F2XX_H$ と $F3XX_H$ は，$XXA0_H$ ～ $XXDF_H$ がひらがなになるかカタカナになるかの違いがある（$F0XX_H$，$F2XX_H$ はカタカナ，$F1XX_H$，$F3XX_H$ がひらがな）。

さらに，$F4XX_H$，$F5XX_H$ には半角外字が割り当てられている。このうち，$F400_H$ ～ $F41F_H$，$F500_H$ ～ $F51F_H$ は“特殊半角外字”とか名づけられているようで，マニュアルによれば，一般のアプリケーションからは使用できないことになっている。

なお，$EC9F_H$ ～ $EDE2_H$ のコードを画面に表示してみると，半角外字が2つ合成されたもの（単純にくっつけた形ではない）が表示される。これはうまくやれば使用できる全角外字の数を増やすことができることを意味する。

$F6XX_H$ ～ $FFXX_H$ は未使用だが，コード入力などを使って無理やり表示すると，半角スペースが表示されるようだ。これらはプログラム中では2バイト半角文字として扱うのが自然だろう。

以上がX68000で使用できる文字のすべてだ。X68000上で動くプログラムを作るからには，これらの文字すべてが正しく扱えるように設計する必要がある。これは僕たちがプログラムを作る上でいつも頭に入れておかなければならないことがらである。

1） 日本のコンピュータで使われている1バイト系文字コードは，正確には“ASCIIコードを拡張してJISで制定したコード”というべきなのだが，以下ASCIIコードで通す。
2） 機種によってはASCIIコード 80_H ～ $9F_H$ や $E0_H$ 以降にセミグラフィック文字が割り当てられている関係で，漢字を使うと使用できない半角文字があったりするのだが，X68000ではもともと漢字の使用が前提であるため，こういった心配はしなくてすむ。
3） 外字はUSKCGM.Xで作る。ちなみに，WP.Xでも同様の範囲が外字になっているが，WP.Xで作った外字はあくまでWP.X内だけで有効であり，システム側の外字とはまったく別の扱いになっている。僕はこれを「外字は使うな」というシステム設計者からのメッセージと受け取った！

BASIC

配列変数を使う

Izumi Daisuke 泉 大介

世の中，変数にひとつだけ値を入れておけばそれで済むほど甘くないんだよ，というわけで配列の登場です。複数のデータを格納しておけるこの便利な変数を使って，今回は素数を求めたり，万年暦を表示するプログラムを作ったりしてみましょう。

先月はX-BASICのプログラムの基本となる関数の作り方を実習しました。関数というのは，原料を与えれば特定のものを作り出してくれる一種のブラックボックスです。そして関数を作るときに使う，「条件が成立している間～を繰り返す」，「条件によって処理を分ける」，「決まった回数だけ処理を繰り返す」といった，プログラムを作るのに便利な命令を紹介しましたね。

今月はこれらの命令を使って実際にプログラムを作りながら，X-BASIC でのプログラミングの方法を紹介していきたいと思います。

便利なデータ集合，配列

「1年は365日あるが，偶数月の日数をすべて加えると何日になるのだろうか」

この問題を考えてみましょう。

まず最初に，どの月には何日あるのかすぐわかるように，

```
tuki1=31
tuki2=28
  ⋮
tuki12=31
```

と12カ月分を変数に格納してみましょう。tuki1は1月を，tuki12は12月を意味しています。問題は偶数月を足した日数を計算することですから，

```
sum=tuki2+tuki4+tuki6……+tuki12
```

とすればいいですね。でもこんなふうに変数を並べていくのは少しマヌケな気がします。

たとえばtukijという変数を用意し，先月やった繰り返しを使って j を2～12まで2ずつ順に変化させ，tuki2～tuki12までを表すことができれば便利そうです。偶数月の日数を合計するプログラムは次のような手順になるでしょう。

```
j=2                ……jに2を入れる
while j <=12       ……jが12以下の間繰り返す
  sum=sum+tukij
  j=j+2
endwhile           ……繰り返すのはここまで
```

最初jは2ですからsumにはtuki2が加えられ，次にjは4になりますからsumにはtuki4が加えられる，という寸法です。

このような処理を実現するためにあるのが配列です。これまで扱ってきた変数にはひとつのデータしか入れることができませんでした。配列は図1のように複数のデータの格納場所を持っている変数です。データの格納場所には順に0，1，2，……と番号が付いています。番号が0から始まることに注意してください。一番左の箱はrei(0)，次の箱はrei(1)というぐあいに，配列は配列名とデータの格納場所をペアにして表します。

通常私たちは数を1から数え始めますが，コンピュータの世界では数を0から数え始めるのが一般的です。このような数え方を「0オリジン」といいます。

X-BASIC の他の変数と同様に，配列も宣言しなければ使えません。配列は，次のようにして宣言します。

```
dim int tuki(12)
```

dimが配列の宣言を表しており，これはtukiという名前の，整数を入れる箱を12番まで用意しなさいという意味になります。箱は0番から始まりますから，tuki配列はtuki(0)からtuki(12)の計13個が用意されます。偶数月の日数をセットするなら，

```
tuki(2)=28
tuki(4)=30
  ⋮
tuki(12)=31
```

と，データを入れる箱の番号と月を一致させておくとわかりやすくていいでしょう。偶数月の日数は，

```
j=2
while j <=12
  sum=sum+tuki(j)
  j=j+2
endwhile
```

で計算できます。さきほどtukijという変数を使って表したものと，ほとんど同じように表現できていますね。

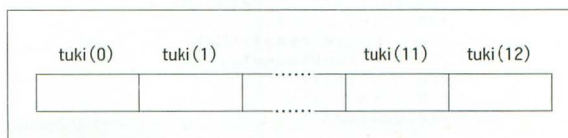
さて，次は月の日数を表す配列に日数をセットします。X-BASIC は配列に簡単に値をセットできるよう次のような代入をサポートしています。

```
tuki={0,31,28,31,……30,31}
```

これでtuki(0)は0に，tuki(1)は31に……，と一気にセットされます（配列の番号と月とを一致させているので，tuki(0)には0を入れておきます）。

X-BASICでは，
〔配列名〕={〔要素並び〕}
という文法で，プログラムのどこででも配列に一挙に値をセットすることができます。この便利な機能をぜひC言語でも採用してほしいものです。

図1 配列のイメージ

tuki(0)	tuki(1)	……	tuki(11)	tuki(12)
		……		

●配列の次元

前で宣言した配列tukiは，横に直線に並んだ箱のイメージでした。これを１次元の配列といいます。図２のように箱を縦横に並べれば，２次元の配列となります。これは，たとえば成績表のように出席番号と科目名の２つによってそこに入るデータが決まるというような場合に便利です。成績表のほかにも，買ったものと買った日付によってデータが決まる家計簿など，この形の配列で処理すると有効なデータは多くあります。

２次元の配列は，

dim int hako2(10, 10)

のように宣言すれば使えるようになります。同様に，

dim int hako3(10, 10, 10)

なら３次元と，次元を増やしていくことができます。クラスの成績表をさらに定期考査の順に並べて処理する場合には，ひとつのデータの格納場所を示すのに出席番号，科目名，定期考査の３つの情報が必要になります。このようにデータ格納場所を示すのに必要な情報の数によって何次元の配列を用意するかが決まるわけです。

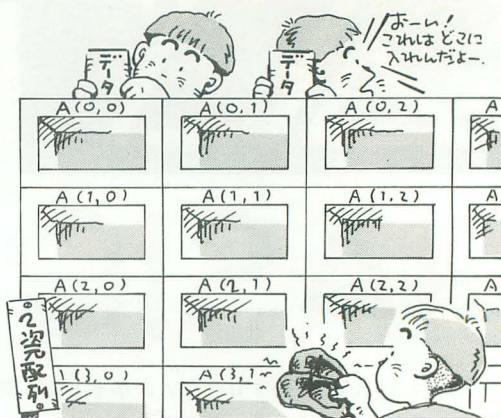
２次元配列は平面のイメージ，３次元配列は直方体のイメージで捕らえることができますが，４次元，５次元となるとちょっとイメージしづらくなりますね。私は，４次元配列は３次元配列が直線に並んだもの，５次元配列は３次元配列が平面に並んだものというイメージで扱っています。

素数を求める

では配列を実際に使ってみることにしましょう。例として取り上げるのは，指定した数以下の素数を列挙するプログラムです。素数を求めるのになぜ配列が必要なのだろうかと不思議に思われるかもしれませんね。実は，「エラトステネスのふるい」というまい求め方があるのです。

図2　2次元配列のイメージ

リスト１　素数を求める

```
 10 /* 素数を求めてみよう
 20 /*
 30 int MAX = 1000
 40 dim int flag(1000)
 50 int i
 60 /*
 70 print 2,
 80 i=3
 90 while i<MAX
100   if flag(i) = 0 then {
110     print i,
120     erase(i)
130   }
140   i=i+2
150 endwhile
160 end
170 /*
180 func erase( base;int )
190   int i = 2
200   while base*i<MAX
210     flag(base*i)=1
220     i=i+1
230   endwhile
240 endfunc
```

エラトステネスのふるいは，「素数は１と自分自身以外に約数を持たない」という性質を利用しています。0～1000までの素数を求めたければ，最初に，

int flag(1000)

として配列を用意します。この配列はflag(0)，flag(1)，……の順に自然数に対応し，最初は０が入っています。対応する数が素数でなければ flag(n) に１を入れることにすると，素数でないものは「私は素数ではありません」とあたかも旗を揚げているように見えますね。flag(フラグ：旗)と命名したのはこのためです。素数かどうか，プログラムを続けるか否か，のように二者択一の問題を扱うときには，旗を揚げているか降ろしているかになぞらえて，こうしたフラグを使うことが少なくありません。

最初flag配列にはすべて０が入っているといいました。つまり，最初の段階ではすべての数を素数だと仮定しているわけです。では最も小さい素数２から始めましょう。２×２以上で1000までのすべての２の倍数ｎに対して，

flag(n)＝1

という操作を行います。つまり，２の倍数は素数ではないということです。

こうして２の倍数をすべて素数の候補から消去したら，次にflag(3)を調べます。flag(3) も０ですから３は素数です。３の倍数を同じように消去します。flag(4) は２の倍数を消去したときに1になっていますから素数ではありません。flag(5) は０です。倍数を消去します。この手順を繰り返していくと，素数だけを残してすべてのflag配列の内容が１になります。そこでflag配列の内容が０のままのものを列挙すれば，1000までの素数が求められるというわけです。

これをプログラムにしたのがリスト１です。ここでは「２以外の偶数は素数ではない」という性質を利用して，flag配列のチェックをflag(3)から，ひとつおきに行うようにしています。つまり奇数しか調べないわけですね。また，素数が求まるたびに画面に表示することにして，実行速度を上げてみました。プログラムはまず30行でMAXに1000をセットしています。40行目でflag配列を宣言していますが，配列の大きさを変数で指定できないので1000と直接数

値で指定しています。

まず，70行で最初の素数である2を画面に表示しておきます。print命令は，

print 1;2;3;4;5

と書くと数を続けて表示します。また，

print 1,2,3,4,5

と書くと，

のように8桁ごとに揃えて数を表示します。70行で使っているのはこのパターンです。

80〜150行がエラトステネスのふるいの部分です。iを3から999まで変化させ，flag(i)が0だったら(iが素数だったら) iを表示し，その倍数をerase関数(180行以降)で消去します。140行でiを2増やしてループを続けます。

万年暦を表示する

今でもあるのかどうかは知りませんが，私が子供の頃，万年暦が付いた学習机がありました。この万年暦は図3のようになっていて，数字の書かれた板の一部が図の四角で囲った場所から見えるようになっています。月初めに数字の書かれた板を左右に動かせば，即座にその月のカレンダーに早変わりする，なかなかよくできたものでした。ただひとつの難点は毎月必ず31日まで表示されていることですが，まァこのくらいは大目に見るとしましょう。

次に挑戦するプログラムは，この万年暦を表示しようというものです。数字の書かれた板の一部が見えるという処理が面倒そうですが，これはX-BASICのmid$関数を使うとうまく処理できるでしょう。

mid$関数は，文字列のn番目の文字からm文字取り出して返す関数です。

たとえば，

print mid$("abcde", 3, 2)

とすると，"abcde"という文字列の3文字目から2文字分取り出すことになりますから，画面には「cd」と表示されます。これを利用して，図3の板を文字列を使って表せば，mid$関数で任意の位置を表示できます。

リスト2を見てください。mondaysという配列を用意し，図3の数字の書かれている板をこのように配列で実現します。表示する月の1日の曜日に合わせて，この一部分をmid$関数で表示するわけです。

10行でstr型の配列を宣言していますが，[ ]でくくられた見慣れないものがありますね。これは，文字列変数に何文字入れることができるようにするかを表しています。これまではこの部分を省略して書いていました。入れることのできる文字数を省略すると，[32]を指定したのと同じことになります。また90行でmonthsというもうひとつの配列を用意し，ここには各月の日数を格納します。

●1日は何曜日？

1日の曜日に合わせてmondays配列の適当な部分を表示するとして，では，その曜日はどのようにすれば知ることができるでしょうか。1週間は7日間ですから，曜日は必ず7日ごとに巡ってきます。ということは，西暦元年の1月1日の曜日と，表示したい月の1日が西暦元年1月1日から数えて何日目なのかがわかれば，曜日を計算により求めることができます。たとえば西暦元年1月1日からの日数を7で割った余りが1なら，その日は西暦元年1月1

						1	2	3	4	5	6	7
2	3	4	5	6	7	8	9	10	11	12	13	14
9	10	11	12	13	14	15	16	17	18	19	20	21
16	17	18	19	20	21	22	23	24	25	26	27	28
23	24	25	26	27	28	29	30	31				
30	31											

図3　机に付いていた万年暦

リスト2　万年暦

```
 10 dim str mondays(5)[40] = {
 20   "                   1  2  3  4  5  6  7",
 30   " 2  3  4  5  6  7  8  9 10 11 12 13 14",
 40   " 9 10 11 12 13 14 15 16 17 18 19 20 21",
 50   "16 17 18 19 20 21 22 23 24 25 26 27 28",
 60   "23 24 25 26 27 28 29 30 31             ",
 70   "30 31                                 "
 80 }
 90 dim int months(12) = { 0, 31, 28, 31, 30, 31, 30,
     31, 31, 30, 31, 30, 31 }
100 /*
110 int year, month, uruu, days
120 str flag
130 int i, goout
140 goout = 0
150 while goout=0
160   input "見たい年は ", year
170   input "見たい月は ", month
180   /*
190   /* 2月の調整
200   /*
210   months(2) = 28
220   if year mod 4 = 0 then months(2) = 29
230   if year mod 100 = 0 then months(2) = 28
240   if year mod 400 = 0 then months(2) = 29
250   /*
260   /* 去年までの閏年の回数
270   /*
280   year = year-1
290   uruu = year/4 - year/100 + year/400
300   months(0) = year*365 + uruu
310   /*
320   /* 先月までの日数
330   /*
340   days = 0
350   for i=0 to month-1
360     days = days+months(i)
370   next
380   days = (days+1) mod 7
390   /*
400   /* カレンダーの表示と終了処理
410   /*
420   print
430   print "日 月 火 水 木 金 土"
440   pr_mondays( days )
450   print
460   print "他の月を見ますか (y/n)";
470   flag = inkey$
480   print
490   if flag = "n" then goout = 1
500 endwhile
510 end
520 /*
530 func pr_mondays( x;int )
540   int i
550   for i=0 to 5
560     print mid$( mondays(i), 19-x*3, 21 )
570   next
580 endfunc
```

日と同じ曜日だといえるわけですね。

1年を365日だとすると，表示したい月までの日数は，

(西暦−1)×365＋「年の始めからの日数」

となります。あとはこれに，(西暦−1) 年までの閏年の回数を加えればOKです。

閏年は次の規則に従って巡ってきます。

1)　西暦が4で割り切れるなら閏年
2)　100で割り切れるなら閏年ではない
3)　400で割り切れるなら閏年

ですから，1988年までの閏年の回数なら，

1988÷4−1988÷100＋1988÷400

とすれば求めることができます。なお，ここでは割り算は整数で行われ，余りは捨てるものとします。

これで万年暦を表示するのに必要な情報はすべて揃いましたね。えっ，西暦元年1月1日の曜日がまだですって？　大丈夫，今日の日付から西暦元年のカレンダーを得ることができます。今日が西暦元年1月1日から何日目なのかを上の式を使って求め，それを7で割った余りを出せばいいのです。もし余りが1なら，西暦元年1月1日は今日と同じ曜日だったはずです。このようにして調べると，西暦元年は月曜日から始まることがわかります。

もちろん，西暦元年から今のものと同じ西暦が使われていたわけではありません。現在使われている西暦は，16世紀に作られたグレゴリオ暦が基本となっています。ですから，西暦元年1月1日の曜日というのは，理論的な値です。

●リスト2を細かく見ると

ではプログラムに戻って説明しましょう。130行まで で変数を宣言し終えたら，次は140，150行です。150～510行はwhile～endwhile のループになっていて，繰り返しいろいろな月を表示できるようにしてあります。この繰り返しを続けるかどうかを決定するための変数がgooutで，gooutが0である間，プログラムは万年暦を表示し続けます。ある月のカレンダーの表示が終わった後，「他の月を見ますか(y/n)」と質問がされ，その答えに応じてgoout を1にするかどうかが決定されます。

160，170行で表示したい年と月を入力してもらい，万年暦のプログラムはスタートです。まず表示する年が閏年かどうかで2月の日数を決めます（220～240行)。次に280，290行で表示する年の前年までの閏年の日数を計算。そして300行で西暦元年1月1日から表示する年の前の年までの日数を求め，第0月の日数としてmonths(0)にセットしています。これは第0月から表示する月の前月までの日数を足せば，西暦元年1月1日からの日数を計算できるようにするためです。この計算を行っているのが350～380行です。390行でそれに1を加えたものを7で割った余りを求め，表示する月の1日の曜日を計算しています。

さぁ次はいよいよカレンダーの表示です。430～460行がカレンダーの表示部分です。「print」とだけ書いてある行は，画面上で改行だけをしなさいという意味になります。何曜日かがわかるように440行で曜日を表示しておいて，実際の表示は pr_mondays関数で行います。この関数の解説は後回しにして，続きを見ていきましょう。

カレンダーを表示し終えたら，470行で続けるかどうかを尋ね，480行で入力を待ちます。500行は続けるかどうかを判定しているところです。押されたキーが「n」だったら，while～endwhile を終了させるためにgooutに1を入れます。gooutが1になっていると，endwhileで150行に戻ったところで条件が偽になりwhile～endwhileは終了します。「n」以外のキーが押されたらgoout は変更しませんので，カレンダーの表示を続けることになります。

残るは540行以降のカレンダー表示部分ですね。関数の引数には，日曜日なら0，月曜日なら1，……というぐあいに，1日が何曜日かという情報が数値で入ってきます。これをもとにmondays配列の表示開始部分を決定し，そこから7日分の21文字を順に画面に出力して終了です。

皆さんの誕生日が何曜日だったのか調べてみてください。そうそう，今年はフランス革命の200周年です。7月14日のバスチーユ牢獄襲撃は何曜日だったのでしょうか。万年暦を操りながら，遠く歴史に想いをはせるのもまた一興。活用してみてくださいね。

プログラム改造のススメ

万年暦をコンピュータで実現してみた今回のプログラムですが，学習机のシステムと同様，2月や4月も31日まで表示されてしまいます。それぞれの月の日数はもう months 配列にセットされていますので，これを使って余分な日は表示しないように変更するのは，X-BASICのいい勉強になるでしょう。

自分が考えた方法でプログラムがうまく動いたときの喜びには，なにものにも代えがたいものがあります。まずはpr_mondays関数を作り直すことから始めてみてはいかがでしょうか。

それでは，また来月お会いしましょう。

清く正しくズリズリと(その2)

Iwai Ippei
満開製作所 祝 一平

先月に引き続き，本気(?)でエディタを作ってしまいましょう講座の第2回目です。今月はカーソル移動からページスクロールへと，次第にサンダーワード0号はその実態を明らかにしていきます。しかし解説を読んでいくと，やはりモットーは「ちまちました気配りと適度の手抜き」にあったよーです。

先月から始めたエディタの制作であるが，2回目の今回はカーソル移動までを仕上げてみるのであった。

で，その前にまことに申し訳ないが，ちょろちょろとバグが出ているので訂正しておくのである。

まずはextern.hであるが，なんとマヌケなことにvalue.cで「UWORD（=unsigned short)」で宣言している変数が，extern.hのなかでは「int」とゆーことになっていた。extern.hを今月掲載のように訂正していただきたい。こーゆーのは致命的なバグで，私はこれに気がつくまでバグが取れずに3時間ほど悩んでしまったのであった。

それから先月号のリスト3，insert_cut_0( )のなかの141行の「w＋9」は，「w＋8」の間違いであった。リスト3の146行と差し換えていただきたい。

あとはgetfile( )であるが，情けないことにここのバグはちょっとひどかった。まるごと今月号のリスト3のように書き換えていただきたい。

カーソル移動を試みるのである

というわけで，今月はカーソル移動関係をサポートしてみた。まだ編集はできないが，テキストを読むことはできるようになったわけである。

カーソル移動というと，「あるだろう」と思われている機能のなかに「スクロール」がある。が，実はこのサンダーワード0号にはスクロールがないのである。「それでエディタができるのかいな？」と思う人もいるだろうが，どっこいちゃんとできてしまうのである。どうしてかというと，行スクロールではなく，ページスクロールで勝負をしているからなのである。これはどういうことかというと，カーソルが一番下にあるとする。このときカーソルをもうひとつ下に行かせると，ED.Xなどの行スクロール型のエディタでは，画面全体が1行上にスクロールし，画面の一番下に新しい行が1行だけ出てくるわけである。

しかしページスクロールではそうならない。カーソルが画面から出ようとしたときには，カーソルのある行が画面の中央付近に来るように画面がガラリと書き換えられるのである。

これは，普通エディタを使うときは編集の対象の行（=カーソルのある行）の前後の内容を見ながら作業を行うという事実から見れば，かなり妥当なスクロールの仕方だと思われる。現実に私はED.Xを使うときには，編集したい行までスクロールして，数行分わざと行き過ぎたあと，目的の行まで戻るようにしている。

そのようなことを最初にいっておいてから，具体的な操作法の説明に移るのであった。

操作方法は，

上 ctrl-P　**下** ctrl-N
左 ctrl-B　**右** ctrl-F
行頭 ctrl-A　**行末** ctrl-E
前ページ ctrl-Z　**次ページ** ctrl-V
先頭 [ESC]+[<]　**最後** [ESC]+[>]

である。

まあ，これぐらいの機能があれば，カーソル移動に関しては問題はないであろう。なお，英文やプログラムを扱うのであれば，[前の単語へ]や[次単語へ]などもあると便利ではあるが，「私はそーゆー機能をあまり使わないもんね」という理由により以上の機能に限定してしまった。ま，それほど問題にはならないであろう(いろいろとやり始めたらキリがないし)。

プログラムの説明である

e.hはまったく追加がない。

extern.hとvalue.cであるが，vz_flag, flush_finishの2変数があり，これはフラグなわけだな。わかっている人はすでにわかっているだろうが，一般的にいうと，プログラミング技術においてフラグとはもっとも安直な手段であり，それゆえに悪い解決法でもある。しかし，強力な手段ということもできる。いい換えれば劇薬だな。まあ，とにかくここではフラグを使っている。使わずに済ませることもできるだろうとは思ったが，使ったほうがプログラムが明解になるだろうと思われたので使ってしまったのである。

それからcctという配列があるが，これはそのときカーソルのある行を「解析」し，テーブルを作るためのものである。なぜこんなものがあるかというと，半角文字が1バイトコードで，全角文字が2バイトコードだからなのである。

このような露骨な配列というのもフラグと同じくらいよくないのであるが，格好をつけるのも面倒なのでそのままにしてある。

で，cctの中身は，

cct［i］［0］　i番目の文字の始まるバイト位置

cct［i］［1］　i番目の文字の表示位置（X座標）

となっている。

たとえば，1行が次のようだったとすると（アルファベットは全部半角），cctは図1のようになるのである。

“AA\t指三本”

このような解析を，現在カーソルのある行に対してしておくのである。

それから，E_flagというフラグもあるが，これは「^E（行末へのジャンプ）」が行われたかどうかのフラグである。このフラグの存在理由は，一度^Eが行われたあと^N，^Pで上下に移動したなら，自動的に行末をたどるようにするためなのである。^Bや^Fなどの，ほかのカーソル移動コマンドがあったならE_flagは0となり，それ以降行末をたどることはしない。これは日本語のテキストを扱っているときは特に必要ないのであるが，Cで書かれたプログラムなど，改行が多い文章の場合は意味が出てくる（これはほかのエディタで便利だったのでパクッてきたのである。よいエディタを目指すには臆面のないパクリも欠かせない）。

変数の説明はこれぐらいにして，次にリスト4である。

関数mainの中でinputという関数を呼び出している。これはFEPなどを使って文字を入力すると，一気に数バイトが流れ込んでくるが，そのとき，それらを「1バイトずつの入力」として見るのと，「まとまりを持った文字列」として見るのとでは，挿入処理の速度が天と地ほども違ってくるのである。今月のようにカーソル移動のコマンドをちまちまと1バイトずつ受け取っている分にはわからないだろうが，本格的に動き出すとこのような処理が不可欠になってくる。一見なんのことはない小技であるが，受身を知らないと致命傷を受けることになる。

で，そのinputであるが，

「キーバッファになにかあるならばどんどん読み出し，なくなったら読み込んだものを入力した文字列として返す」関数なのである。念のためにいっておくが，「1行入力」では［CR］を押さなければならないのでだめなのである。

で，先ほども述べたinsert_cut0のバグであるが，タブを考慮するときに，

△＝(△＋9)　&　0xfff8;

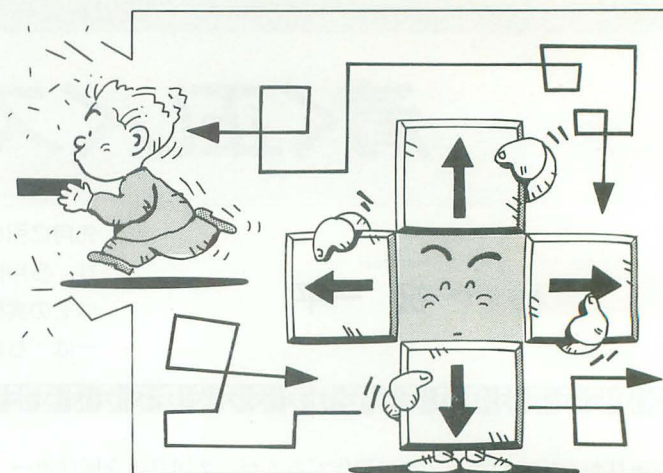
としていたのであるが，これはとんでもないオオボケであった。これは，

図1　cctの内容

たとえば“AAA\t指本三”を解析すると

	0	1	2	3	4	5	6	7
バイト位置	0	1	2	3	4	6	8	10
X座標位置	0	1	2	3	8	10	12	14
	A	A	A	\t (タブ)	指	本	三	\0 (NULL)

そしてcklen==8となっている

△＝(△＋8)　&　0xfff8;

でなくてはいけなかったのである。

さて，今月の中で一番わかりずらいのがflush_finishであろう。プログラムでは，画面書き換えの最中に^V，^Zを検知したならば，「まだ終わってないよ」というフラグ（つまりそれがflush_finish）を立てて（＝1にして）返ってくるようにしている。で，呼び出した側ではそれを検知して，「それじゃもう1回画面書き換えをやってきてね」という次第なのである。ただしこれは^V，^Zの処理中だけのことなのである。そのことがわかるようにvz_flagというフラグを使っている（フラグの2段攻撃である）。そこらへんのことは関数vz_interruput( )でチェックしている。

なぜこんな変なことになっているかというと，これは「改ページの速度を上げるため」なのである。こうすることによって，不必要な画面書き換えを省けるのである。

たとえば，素早く2度^Vを押したならば，まず最初の^Vに反応して画面書き換えが始まるわけである。ところがその作業をしている途中で，vz_interruput( )によって，「また^Vが押されたよーん」ということがわかるわけだな（プログラムをよく見てね）。と，いうことになれば，もうこれ以上画面の書き換えを続けても無駄になってしまうわけだ。そこで画面書き換えをさっさとやめて「途中でやめてきたよ」といいつつ（フラグを立てて）戻るわけである。そうすると，呼び出した側では，「途中で返って来た(flush_finish==0)ということは，ははぁんvz_interruput( )だナ」ということになり，「それじゃもう一度［改ページ］をして，もっと先のところを画面に表示してきなさい」ということになるのである。つまり手際よく余計な作業を省いたわけである。

よいエディタを作るには，ここらへんのちまちました気配りも必要なのである。ああしんど。

cursor( )

これは今回中心的な関数である。

とはいっても機能は単純で，現在行(cl)からccsにコピーし，そのデータを解析してテーブルを作り，locateでカーソル位置を指定するだけである。これは上下左右のカーソル移動の直後などに頻繁に呼び出される関数である。

nbefore(l, c)

これは，指定した行(l)からc行さかのぼった行の番号を返す

関数である。それだけだったら単純でよいのであるが，c行さかのぼる前に先頭行に行き着いたなら，そこまでたどった行数を返すことになっている。ただし「たどった行数」をそのまま返したのでは，途中で先頭行に達したのかどうかが判別できないので，その場合はセコく「−たどった行数−1」を返すことになっている。なんで「〜−1」かというと，そうしておかないと「0」ということがあり得るから，そうしたら区別がつかないからなのだね。なんともちまちまとしたキタナイ話であるが仕方がない。

で，nbefore( )の値を受ける場合は，符号のある型（intなど）でなくてはならないことに注意。

nnext(l, c)

nbefore( )の逆である。

来月は編集ができたりする

まあ，ざっとこれぐらいであろうか。細かく説明していくときりがないのでここらへんでやめておく。例によってプログラム中に注釈を入れておいたので読んでいただきたい。alpha.cのほうはまさしく対症療法の山であるが，これはcursor( )，cursor0( )を有効に呼び出すことによってこの程度の大きさに抑えることができたのである。

てなとこで，今月はこれまでである。来月はいよいよ編集（挿入/削除）をできるようにする予定である。それでは，ごきげんよう，さようなら。

リスト1　E.H（先月号発表分と組み合わせてコンパイルのこと。そのときLINE.Cを忘れずに）

```
==================  E.H  ==================
        1: /* 第１回目↓ */
        2:
        3: #define Xwidth          80      /* 桁数：最高９６ */
        4: #define Ywidth          20      /* 行数：最高３２ */
        5: #define MaxTextLine     200     /* テキストの最大行数 */
        6: #define MAXLINE         100     /* 作業用の文字列の長さ */
        7: #define Kaigyou         '▼'     /* 改行マーク 81A5H */
        8:
        9: #define txt(X)          (buff[(X)].data)        /* 文字列データ本体 */
       10: #define before(X)       (buff[(X)].mae)         /* 前の行番号 */
       11: #define next(X)         (buff[(X)].ato)         /* 後の行番号 */
       12: #define EOL             (0xffff)                /* 行の終端：ありえない行番号 */
       13:
       14: typedef struct LINE {
       15:         UBYTE data[Xwidth+1];   /* 1 line and EOS */
       16:         UWORD mae;              /* 前の行番号 */
       17:         UWORD ato;              /* 後ろの行番号 */
       18: };
       19:
       20: /* 第１回目↑ */
       21:
       22: /* 追加はなしである */
       23:
       24: /* 第２回目↑ */
       25:
```

リスト2　EXTERN.H

```
==================  EXTERN.H  ==================
        1: extern struct LINE buff[MaxTextLine];    /* データ本体 */
        2:
        3: extern FILE *fp0;                         /* ファイル構造体へのポインタ */
        4: extern UBYTE filename0[MAXLINE];          /* ファイル名 */
        5:
        6: extern int cx,cy;           /* カーソル位置 */
        7: extern int cxk; /* その行の何文字目にいるか（全角／半角／タブがあるから） */
        8:
        9: extern UWORD fl;                /* first line リストの先頭の行番号 */
       10: extern UWORD bl;                /* bottom line リストの最後の行番号 */
       11:
       12: extern UWORD hl;                /* home line 画面一番上の行番号 */
       13: extern UWORD cl;                /* current line 現在編集中の行番号 */
       14:
       15: extern UWORD free_line;         /* 未使用の行リストの先頭 */
       16: extern int full_flag;           /* 空行がなくなった時用のフラグ */
       17:
       18: extern UWORD kinsetzen[];       /* 禁則文字のセット */
       19:
       20: /* 第１回目↑ */
       21:
       22:
       23: extern UBYTE com;           /* コマンドコード格納用 */
       24:
       25: extern int vz_flag;             /* ^Z,^Vを実行中のフラグ（連続改ページに対応するため） */
       26: extern int flush_finish;        /* ^Z,^Vを完了したかのフラグ（同上） */
       27:
       28: extern char vz_control[];
       29:                             /* ^V,^Z のコントロールコードを持っている(あまりよくない) */
       30:
       31: extern UBYTE ccc;           /* カーソル位置のキャラクタ */
       32: extern UWORD ccode;         /* カーソル位置のキャラクタ（全角の場合あり） */
       33:
       34: extern int cklen;           /* 現在行に、全角文字、半角文字合わせて何文字あるか */
       35:
       36: extern UBYTE ccs[MAXLINE];      /* 現在行をコピーしておく*/
       37: extern UBYTE cct[MAXLINE][2];   /* 現在行の解析結果のテーブル */
       38:
       39: extern int E_flag;          /* ^E（行末へのジャンプ）が実行されたかどうかのフラグ */
       40:
       41: /* 第２回目↑ */
```

リスト3　A2.C

```
==================  A2.C  ==================
     1: #include        <stdio.h>
     2: #include        <basic0.h>
     3: #include        <iocslib.h>
     4: #include        <conio.h>
     5: #include        <class.h>
     6: #include        "e.h"
     7: #include        "extern.h"
     8:
     9: main(argc,argv)
    10: int argc;
    11: UBYTE *argv[];
    12: {
    13:         int i,w;
    14:         UBYTE coms[MAXLINE];
    15:
    16:         init();
    17:         if (argc < 2) strcpy(filename0,"DELETE.ME");     /* ファイル名の指定がない */
    18:         else strcpy(filename0,argv[1]);
    19:         fp0 = fopen(filename0,"r+t");                    /* ファイル更新でオープンしてみる */
    20:         if (fp0 == NULL) fp0 = fopen(filename0,"w+t");  /* 新しくファイルを作る */
    21:         if (fp0 == NULL) error("main:file open");        /* なんか変だぞ */
    22:
    23:         getfile(fp0);           /* ファイルの読み込み */
    24:
    25:         cl = cxk = cx = cy = 0; /* カーソル位置などの設定 */
    26:         flush(0);               /* 位置画面表示 */
    27:
    28:         cursor();
    29:         while(1) {
    30:                 if (input(coms,MAXLINE) == 1) {
    31:                         com = *coms;
    32:                 } else {        /* FEPからの入力 */
    33: /*                      n_insert(coms);         まだだよーん */
    34:                         cursor();
    35:                         continue;
    36:                 }
    37:                 if (0x20 <= com) {      /* 1文字入力した */
    38: /*                      insert(com);            まだだよーん */
    39:                         continue;
    40:                 }
    41:                 switch(com + 0x40) {
    42:                 case 'A':       /* 行頭へ */
    43:                         do_A();
    44:                         E_flag = 0;     break;
    45:                 case 'B':       /* 左 */
    46:                         do_B();
    47:                         E_flag = 0;     break;
    48:                 case 'E':       /* 行末 */
    49:                         do_E();
    50:                         E_flag = 1;     break;
    51:                 case 'F':       /* 右 */
    52:                         do_F();
    53:                         E_flag = 0;     break;
    54:                 case 'L':       /* 書き直し */
    55:                         do_L(); break;
    56:                 case 'N':       /* 下 */
    57:                         do_N(); break;
    58:                 case 'P':       /* 上 */
    59:                         do_P(); break;
    60:                 case 'V':       /* 次頁 */
    61:                         do_V();
    62:                         E_flag = 0;     break;
    63:                 case 'X':       /* ^X */
    64:                         do_X(); break;
    65:                 case 'Z':       /* 前頁 */
    66:                         do_Z();
    67:                         E_flag = 0;     break;
    68:                 case '[':       /* ESC */
    69:                         do_ESC();       break;
    70:                 }
    71:                 cursor();       /* 現在行を解析しておく */
    72:         }
    73:         finis();                /* コンソールを元に戻したりする */
    74: }
    75:
    76: init()
    77: {
    78: /*      ◆◆◆◆          この部分は先月と同じ      ◆◆◆      */
    79:
    80:         E_flag = 0;
    81: }
    82:
    83: finis()
    84: {
    85: /*      ◆◆◆◆          この部分は先月と同じ      ◆◆◆      */
    86: }
    87:
    88: /* ファイルを読み込む */
    89: getfile(fp)
    90: FILE *fp;
    91: {
    92:         int wl,wl0,c;
    93:         UBYTE l[MAXLINE*2];     /* MAXLINE > Xwidth */
    94:         UBYTE s[MAXLINE*2];
    95:
    96: /* ◆◆◆ 次の1行訂正 ◆◆◆ */
```


▶ぼくの学校の文化祭のテーマはなんと「夢をかたちに」です。くそー，来年は実行委員になって「夢のつづきを語ろう」にしてやるぜ！　しまった，俺は3年生だった。
庄子　真海（15）宮城県

```
 97:         fl = wl = newline(EOL,EOL);                    /* 最初に1行分を手に入れる */
 98:         under_print("ファイル読み込み中…");     /* 時間の掛かる作業の時はメッセージを出す */
 99:
100:         *s = '¥0';                          /* 前からの繰り越しはなし */
101:         while(! feof(fp)) {
102:                 fgets(l,MAXLINE,fp);    /* 適当な長さだけ読み出す */
103:                 c = strlen(l);
104:                 if (! c) break;         /* ◆◆◆ 追加 ◆◆◆ */
105:
106:                 if (l[(c = strlen(l) - 1)] == '¥n') {   /* 最後が改行なら'▼'にする */
107:                         l[c++] = Kaigyou >> 8;
108:                         l[c++] = Kaigyou & 0xff;
109:                         l[c] = '¥0';                    /* 文字列終了も忘れずに */
110:                 }
111:                 strcat(s,l);            /* 前からの繰り越しの後に加える */
112:         /* 注：MAXLINE > Xwidthだから、sの中には必ず1行分以上の文字列が入っている */
113:
114:                 do {
115:                         insert_cut0(s,l);       /* 一行分を切り分ける */
116:                         strcpy(buff[wl].data,s);        /* 納める */
117:                         wl = getnext(wl0 = wl); /* 次の行番号を取ってくる */
118:                         if (wl == EOL) {        /* もう空行がない！ */
119:                                 bl = wl0;
120:                                 under_blanc();
121:                                 return;
122:                         }
123:                         strcpy(s,l);            /* sに余りを転送 */
124:                 } while(strlen(s) >= Xwidth);   /* sが十分短くなったら新しく読み出して来る */
125:         }
126:
127:                 /* ◆◆◆ 追加 ◆◆◆ */
128:         do {                                /* うーむ、無駄無駄 */
129:                 insert_cut0(s,l);           /* 一行分を切り分ける */
130:                 strcpy(buff[wl].data,s);        /* 納める */
131:                 wl = getnext(wl0 = wl); /* 次の行番号を取ってくる */
132:                 if (wl == EOL) {            /* もう空行がない！ */
133:                         bl = wl0;
134:                         under_blanc();
135:                         return;
136:                 }
137:                 strcpy(s,l);                /* sに余りを転送 */
138:         } while(strlen(s) > 0);             /* sが十分短くなったら新しく読み出して来る */
139:                 /* ◆◆◆ 追加 ◆◆◆ */
140:         bl = wl;        /* 最後の行番号をblに覚えておく */
141:         under_blanc();
142: }
143:
144: insert_cut0(s,t)
145: /* ◆◆◆ 次の1行訂正 ◆◆◆ */
146:                         w = (w+8) & 0xfff8;     /* !!! */
147:
148:
149: /* 指定行から下を書き直す */
150: flush0(y,l)
151: int y;          /* 書き直し開始位置 */
152: UWORD l;        /* 書き直し開始行：最終行（bl）ではないこと */
153: {
154:         flush_finish = 0;       /* そのうちわかる */
155:         for(;y < Ywidth-3;y++) {
156:                 flush_1line(y,txt(l));  /* 1行表示 */
157:                 l = next(l);
158:                 if (l == EOL) {         /* 最終行に突き当たった */
159:                         B_PRINT("¥x1b[0J");     /*  カーソル位置から最後までクリア */
160:                         break;
161:                 }
162:
163: /* ◆◆◆ 次の3行訂正 ◆◆◆ */
164:                 if (vz_interrupt()) {   /* コマンドの連発？ */
165:                         return;
166:                 }
167:                                 /* そのうちわかる */
168:         }
169:         flush_finish = 1;       /* そのうちわかる */
170: }
171:
172: /* コマンド／文字の入力 */
173: /* ＦＥＰからの入力に対応できるように、何かキー入力があるうちはどんどこと取り込んでいく */
174: /* 原則として文字列を返す（もちろん1文字しか返さない時もあるよ */
175: input(s,max)
176: char *s;
177: {
178:         int c;
179:
180:         c = 0;
181:         while(c == 0) {         /* 必ず何か1文字は入力する */
182:                 while(kbhit() && (c < max-1)) { /* ありったけ読み出す */
183:                         s[c++] = INKEY();
184:                 }
185:         }
186:         s[c] = '¥0';    /* 文字列の終端記号を忘れずに */
187:         return(c);
188: }
189:
190: /* ctrl-X + ? の処理 */
191: do_X()
192: {
193:         under_print("CTR-X:");  /* モードを示す */
194:         com = INKEY();          /* 1文字だけ入力 */
195:         under_blanc();          /* モード表示を消す */
196:         cursor();
```

▶うおーっ，やったー「サンダーフォースII」をといたずぉーっ，と喜んでいた次の日，学校の帰りに自転車でコケて指の先の皮がズズッとむけてしまい(これがまた痛いんだ)。指用のバンソーコーが必需品となってしまった。ああ，ゲームができないよぉー！

清水　了 (15) 大阪府

```
197:            switch(com+0x40) {
198:            case 'C':       /* 終了 */
199:                    finis();
200:            case 'D':       /* コア・ダンプ */
201:                    under_print("コア・ダンプ中…");         /* メッセージ表示 */
202:                    core_dump();
203:                    under_blanc();                          /* メッセージを消す */
204:                    break;
205:            case 'L':       /* 表示位置を変えずに画面書き直し */
206:                    flush(hl);
207:                    break;
208:            case 'S':       /* セーブ */
209:            /* まだだよーん */
210:                    break;
211:            case 'G':       /* 中断 */
212:                    bell();
213:                    break;
214:            default:
215:                    break;
216:            }
217:            key_empty();    /* key buff empty */
218: }
219:
220: /* ESC + ? の処理 */
221: do_ESC()
222: {
223:            under_print("ESC:");    /* モードを示す */
224:            com = INKEY();          /* 1文字だけ入力 */
225:            under_blanc();          /* モード表示を消す */
226:            switch(com) {
227:            case '<':       /* テキストの先頭にジャンプ */
228:                    cl = fl;
229:                    cxk = cx = cy = 0;
230:                    flush(fl);
231:                    break;
232:            case '>':       /* テキストの最終にジャンプ */
233: /*                 insert_cut_all();       あとでね */
234:                    cl = bl;
235:                    do_E();         /* 行末に行く:cx,cxkをセットする */
236:                    cy = center_flush();
237:                    break;
238:            case '¥x20':    /* マークセット */
239: /*                 まだだよーん    */
240:                    break;
241:            case '¥007':    /* 中断 */
242:                    bell();
243:                    break;
244:            default:
245:                    break;
246:            }
247:            key_empty();    /* key buff empty */
248: }
249:
```

リスト4 SUB.C

```
==================  SUB.C  ==================
   1: /*       ◆◆◆◆          ここまでは先月と同じ      ◆◆◆     */
   2:
   3: /* カーソル位置を設定し、現在行の解析をする */
   4: cursor()          /* set ccc & ccs */
   5: {
   6:         locate(cx,cy);
   7:         cursor0();
   8: }
   9:
  10: /* 現在行の解析をする */
  11: cursor0()         /* set ccc & ccs */
  12: {
  13:         strcpy(ccs,buff[cl].data);      /* 現在行をccsにコピーする */
  14:         ktable();                       /* 解析する */
  15:         ccode = ccc = ccs[cct[cxk][0]]; /* カーソル位置のキャラクタを取り出す */
  16:         if (iskanji(ccc)) ccode = (ccc << 8) | ccs[cct[cxk][0]+1];
  17:                                         /* 全角文字への対応 */
  18: }
  19:
  20: /* l行目からc行遡る。c行の前に先頭行に達したら(-遡った行数-1)を返す */
  21: nbefore(l,c)
  22: int l,c;
  23: {
  24:         int i;
  25:
  26:         for(i = 0;i < c;i++) {
  27:                 if (l == fl) {  /* 先頭行に達した */
  28:                         return(-i-1);
  29:                 }
  30:                 l = before(l);  /* 前の行をたぐる */
  31:         }
  32:         return(l);
  33: }
  34:
  35: /* l行目からc行下る。c行の前に最終行に達したら(-下った行数-1)を返す */
  36: nnext(l,c)
  37: int l,c;
  38: {
  39:         int i;
```


▶インディ・ジョーンズは凄い。この映画には活字や絵じゃ表せない映像の素晴らしさがあふれている。ラストシーンで泣き笑いになったところにあのテーマが流れてきて……。幸せな一日でした。　晴山　佳彦(19)千葉県

```
 40:
 41:         for(i = 0;i < c;i++) {
 42:                 if (l == bl) {  /* 最終行に達した */
 43:                         return(-i-1);
 44:                 }
 45:                 l = next(l);    /* 次の行をたぐる */
 46:         }
 47:         return(l);
 48: }
 49:
 50: /* clが画面の中央に来るように画面を書き変える */
 51: /* clのY座標を返す */
 52: center_flush()
 53: {
 54:         int l;
 55:
 56:         l = nbefore(cl,((Ywidth-4) >> 1));
 57:         if (l < 0) {    /* 指定数だけ遡る前に先頭行に出くわした。-l-1が遡った行数 */
 58:                 flush(fl);      /* 先頭行を一番上にして書き直し */
 59:                 return(-l-1);                   /* clのY座標を返す */
 60:         } else {        /* 指定行数だけ遡った所の行番号はl */
 61:                 flush(l);       /* lを一番上にして書き直し */
 62:                 return((Ywidth-4) >> 1);        /* clのY座標を返す */
 63:         }
 64: }
 65:
 66: /* ccsを分解してcctにテーブルを作る */
 67: ktable()
 68: {
 69:         int c,x,i;
 70:         UBYTE w;
 71:
 72:         i = c = x = 0;
 73:         while(1) {
 74:                 cct[i][0] = c;          /* バイト位置をセット */
 75:                 cct[i++][1] = x;        /* x座標位置をセット */
 76:                 if (iskanji(w = ccs[c++])) {    /* 全角文字の場合 */
 77:                         c++;
 78:                         x += 2;
 79:                 } else if (w == '¥t') {         /* タブの場合 */
 80:                         x = (x+8) & 0xfff8;
 81:                 } else if (w) {                 /* 普通の半角文字 */
 82:                         x++;
 83:                 } else {
 84:                         cklen = i;              /* 文字数を覚えておく */
 85:                         break;  /* return */
 86:                 }
 87:         }
 88: }
 89:
 90: /* ^V,^Zでインタラプトを起こすかどうか */
 91: vz_interrupt()
 92: {
 93:         unsigned int c;
 94:
 95:         if (vz_flag && kbhit()) {
 96:                 if ((c = INKEY()) == vz_flag)
 97:                         return(1);      /* 立て続けに改ページコマンドが入力された */
 98:         }
 99:         return(0);
100: }
```

リスト5 VALUE.C

```
==================  VALUE.C  ==================
     1: #include        <stdio.h>
     2: #include        <basic0.h>
     3: #include        <iocslib.h>
     4: #include        <conio.h>
     5: #include        <class.h>
     6: #include        "e.h"
     7:
     8: struct LINE buff[MaxTextLine];  /* データ本体 */
     9:
    10: FILE *fp0;                              /* ファイル構造体へのポインタ */
    11: UBYTE filename0[MAXLINE];               /* ファイル名 */
    12:
    13: int cx,cy;              /* カーソル位置 */
    14: int cxk;                /* その行の何文字目にいるか(全角/半角/タブがあるから) */
    15:
    16: UWORD fl;                       /* first line リストの先頭の行番号 */
    17: UWORD bl;                       /* bottom line リストの最後の行番号 */
    18:
    19: UWORD hl;                       /* home line 画面一番上の行番号 */
    20: UWORD cl;                       /* current line 現在編集中の行番号 */
    21:
    22: UWORD free_line;                        /* 未使用の行リストの先頭 */
    23: int full_flag = 0;              /* 空行がなくなった時用のフラグ */
    24:
    25: UWORD kinsetzen[] = {   /* 禁則文字のセット */
    26: '、', '。', '?', '!',
    27: '」', '』', ')', '}', ']', '"',
    28: 'あ', 'い', 'う', 'え', 'お',
    29: 'ア', 'イ', 'ウ', 'エ', 'オ',
    30: 'っ', 'ッ',
    31: 'ゃ', 'ゅ', 'ょ',
```

▶ハードディスクがほしい。でも高くて買えない。X68000ノーマルを50万近くで買った私としては,いまのACE, EXPERT, PROのHDを持っている人がうらやましい。

石堂　直樹 (19) 愛知県

```
32: 'ャ', 'ュ', 'ョ',
33: '▼',0
34: };
35:
36: /* 第1回目↑ */
37:
38:
39: UBYTE com;      /* コマンドコード格納用 */
40:
41: int vz_flag = 0;        /* ^Z,^Vを実行中のフラグ(連続改ページに対応するため) */
42: int flush_finish;       /* ^Z,^Vを完了したかのフラグ(同上) */
43:
44: char vz_control[] = "¥x16¥x1a";
45:                 /* ^V,^Z のコントロールコードを持っている(あまりよくない) */
46:
47: UBYTE ccc;      /* カーソル位置のキャラクタ */
48: UWORD ccode;    /* カーソル位置のキャラクタ(全角の場合あり) */
49:
50: int cklen;      /* 現在行に、全角文字、半角文字合わせて何文字あるか */
51:
52: UBYTE ccs[MAXLINE];     /* 現在行をコピーしておく*/
53: UBYTE cct[MAXLINE][2];  /* 現在行の解析結果のテーブル */
54:
55: int E_flag;     /* ^E(行末へのジャンプ)が実行されたかどうかのフラグ */
56:
57: /* 第2回目↑ */
```

リスト6 ALPHA.C

```
==================  ALPHA.C  ==================
    1: #include         <stdio.h>
    2: #include         <basic0.h>
    3: #include         <iocslib.h>
    4: #include         <conio.h>
    5: #include         <class.h>
    6: #include         "e.h"
    7: #include         "extern.h"
    8:
    9: /* 行頭へ */
   10: do_A()
   11: {
   12:         cx = cxk = 0;   /* このあとでcursor()が実行されるからこれでいいのだ */
   13: }
   14:
   15: /* 左 */
   16: do_B()
   17: {
   18:         if (cx == 0) {                  /* 行頭にいる場合 */
   19:                 if (cl == fl) return;   /* 先頭行ならなにもしない */
   20:                 cl = before(cl);        /* 前の行に行く */
   21:                 do_E();                 /* 前の行の行末に行く */
   22:                 if (cy) {               /* 一番上の行ではなかったなら */
   23:                         cy--;           /* 一行上に上がるだけ */
   24:                 } else {                /* 一番上の行だったなら */
   25:                         cy = center_flush();    /* ページの書き換え */
   26:                 }
   27:         } else {                        /* 行頭でないのなら */
   28:                 cxk--;                  /* 1文字前へ */
   29:                 cx = cct[cxk][1];       /* x座標をそろえる */
   30:         }
   31: }
   32:
   33: /* 行末 */
   34: do_E()
   35: {
   36:         cursor0();              /* 解析をし直す(他から呼ばれることがあるから) */
   37:         cxk = cklen-2;          /* 行末の文字へ */
   38:         cx = cct[cxk][1];       /* 行末の文字の上へ */
   39:         cursor0();              /* ccodeを得るためにもう一度解析し直す */
   40:         if ((ccode == Kaigyou)                  /* カーソルの下は改行マーク? */
   41:          || (cct[cxk+1][1] >= Xwidth)) {        /* 1文字右に行くとXwidth以上? */
   42:         } else {                        /* そうでないのなら */
   43:                 cxk++;
   44:                 cx = cct[cxk][1];       /* もうひとつ右へ */
   45:         }
   46: }
   47:
   48: /* 右 */
   49: do_F()
   50: {
   51:         if ((cl == bl) && (ccode == '¥0')) return;
   52: /*
   53: 次の行に移る条件:
   54: (1)下に文字がない状態の時に^Fが入力された
   55: (2)下に改行マーク(▼)がある時に^Fが入力された
   56: (3)1文字右に行くとXwidth以上になってしまう
   57: */
   58:         if ((cxk > cklen-2)     /* カーソルの下に文字がない */
   59:          || (ccode == Kaigyou)  /* カーソルの下は改行マーク */
   60:          || (cct[cxk+1][1] >= Xwidth)) {        /* 1文字右に行くとXwidth以上 */
   61:                 cl = next(cl);
   62:                 cxk = cx = 0;
   63:                 cy++;
   64:                 if (cy >= Ywidth-3) {           /* 次の頁へ */
   65:                         cy = center_flush();    /* cl行を中心にしてflushする */
   66:                 }
   67:         } else {
```

▶最近，記事内容が充実してきた。加えて3DやCG特集である。感涙の今日このごろである。
菊田 俊彦（31）茨城県

```
 68:                cxk++;                          /* 単純に右へ行く */
 69:                cx = cct[cxk][1];
 70:        }
 71: }
 72:
 73: /* 現在行を中心にして書き直し */
 74: do_L()
 75: {
 76:        cy = center_flush();
 77: }
 78:
 79: /* 下 */
 80: do_N()
 81: {
 82:        UWORD cl0;
 83:
 84:        if (cl == bl) return;   /* 最終行ならなにもしない */
 85:        cl0 = next(cl);         /* 次の行をチェック */
 86:        if ((strlen(buff[cl0].data) == 0)                  /* 次の行が0文字で */
 87:         && (! jstrchr(buff[cl].data,Kaigyou))) {          /* 現在行に改行がないなら */
 88:                return;                                    /* 何もせずに帰る */
 89:        }
 90:
 91:        cl = cl0;           /* 次の行を現在行にする */
 92:        cursor0();          /* 解析しておく */
 93:        if (E_flag) {       /* 右端にいた */
 94:                do_E(); /* それなら右端に行く */
 95:        } else {            /* さもなくば */
 96:                if (cklen-2 < cxk) cxk = cklen-2;          /* 新しい行が短いなら行末 */
 97:                cx = cct[cxk][1];          /* 文字の“番目”を変えない */
 98:        }
 99:        cy++;               /* Y座標を増やす */
100:        if (cy >= Ywidth-3) {              /* 最下行を超えたら次頁 */
101:                cy = center_flush();       /* cl行を中心にしてflushする */
102:        }
103: }
104:
105: /* 上 */
106: do_P()
107: {
108:        if (cl == fl) return;   /* 先頭行ならなにもしない */
109:        cl = before(cl);        /* 前の行をたぐる */
110:        cursor0();              /* 解析しておく */
111:        if (E_flag) {   /* 右端にいた */
112:                do_E(); /* それなら右端に行く */
113:        } else {        /* さもなくば */
114:                if (cklen-2 < cxk) cxk = cklen-2;          /* 新しい行が短いなら行末 */
115:                cx = cct[cxk][1];          /* 文字の“番目”を変えない */
116:        }
117:        cy--;           /* Y座標を減らす */
118:        if (cy < 0) {   /* 最上行を超えたら前頁 */
119:                cy = center_flush();       /* cl行を中心にしてflushする */
120:        }
121: }
122:
123: /* 次頁 */
124: do_V()
125: {
126:        int w;
127:
128: /*     insert_cut_all();          まだだよーん */.
129:        cursor0();                 /* 念のため解析しておく */
130:        if ((cl == bl) && (ccode == '¥0')) return; /* 最後端なので何もしない */
131:        vz_flag = 0x16;            /* vz_flag をセットする */
132:        do {
133:                if ((w = nnext(hl,Ywidth-5+((Ywidth-4) >> 1))) < 0) {
134:                        cl = bl;           /* 最終行に達した */
135:                } else {
136:                        cl = w;
137:                }
138:                cy = center_flush();
139:                cxk = cx = 0;              /* 行頭へ */
140:        } while(! flush_finish);           /* vz_interrupt()にかかったならもう一度 */
141:        vz_flag = 0;               /* vz_flag をクリアする */
142: }
143:
144: /* 前頁 */
145: do_Z()
146: {
147:        int w;
148:
149:        if (hl == fl) {            /* 既に先頭頁に達している */
150:                cxk = cx = cy = 0;
151:                cl = fl;           /* 最上行の行頭へ */
152:                return;
153:        }
154:        vz_flag = 0x1a;            /* vz_flag をセットする */
155:        do {
156:                if ((w = nbefore(hl,Ywidth-5-((Ywidth-4) >> 1))) < 0) {
157:                        cl = fl;           /* 先頭行に達した */
158:                } else {
159:                        cl = w;
160:                }
161:                cy = center_flush();
162:                cxk = cx = 0;              /* 行頭へ */
163:        } while(! flush_finish);           /* vz_interrupt()にかかったならもう一度 */
164:        vz_flag = 0;               /* vz_flag をクリアする */
165: }
```

▶宇野総理の退陣は，「SUPER大戦略」で負けそうになったとき，慌ててリセットするのと似てる，と思う。　山下　智也（19）大阪府

宇宙要塞CADを攻略せよ!

プロジェクトチームDōGA かまた ゆたか
MAX 田口

今回はDōGA・CGAシステム最大の難関，難攻不落の宇宙要塞 CAD（3D サーフェスモデリングツール）を攻略してみようと思います。そのため，執筆陣としても，マニュアルの第3章「CGA制作入門」の作者として有名な(？)MAX田口先生もお呼びしました。

CGAシステムにたくさんのご応募誠にありがとうございます。おかげさまで，8月1日発送分はあっという間になくなってしまいました。また，カンパもスタッフの予想をはるかに上回るもので，一時は実費の増大によって赤字の噂もあった収支も，ちゃんと黒字になるメドが立ちました。皆さんの期待の大きさを改めて痛感し，身の引き締まる思いです。なお，まだお手元にCGAシステムが届いていない方は，インフォメーションのコーナーをご覧ください。

とまあ，堅苦しいあいさつはこのくらいにして，今回のメインテーマであるCADの説明に移ります。CADは，物体の形状を3面図と透視図を見ながらマウスでデザインしていくプログラムです。

物体形状デザイン（モデリング）という作業は，CGA制作の出発点でありながら，直感的に3Dをとらえるセンスやコツコツ積み上げていく忍耐力を必要としますので，「誰にでも，すぐ簡単に」というわけにはなかなかいきません。また，このCADは本格的な使用に耐えるだけの高い機能を持っているだけに，かなり複雑で挫折してしまう初心者も少なくありません。このあたりに難攻不落の宇宙要塞と呼ばれる所以があるのです。

図1　画面に向かって左上が平面図(上がY軸の正方向，右がX軸の正方向)，左下が側面図(上がZ軸の正方向，右がX軸の正方向)，右下が正面図(上がZ軸の正方向，右がY軸の正方向)，右上が透視図です。操作図面は図面の枠が現在のペンカラーの破線により囲まれて表示されます。各図面は，その図面のまわりにある矢印をクリックすることで表示領域を変えることができます。右端の操作パネルは上から順に，Aパネル，Bパネル，Cパネルと言います。Cパネルは，マウスをクリックすると，Panel 1からPanel 5に順に変わっていきます。

まずはMAX田口先生に「CAD入門編」をお願いしたあと，再び私がCADを使いこなすテクニックをいくつかご紹介することにします。

MAX田口のCAD入門

●まずは面画の説明から

私は，今回の宇宙要塞CAD攻略隊の隊長に選ばれたMAX田口である。与えられたページは少ない。しかも攻略隊のメンバーは初心者ばかりである。はなはだ心もとないが，CADにも弱点はあるはずだ。進め，勇者たちよ！CAD攻略なくして，作品は作れないのだ！

さて，図1が宇宙要塞CADの見取図（画面）である。3つのパネルのうち，Aパネルはモードの選択や一般のコマンド，Bパネルは面ポインタと3Dカーソルの移動や操作，Cパネルは表示画面の設定やその他の環境設定に関係したコマンドがそれぞれまとめてある。しかもCパネルは，さらにPanel 1からPanel 5に分かれている。もちろん各パネルの詳しい説明は，我らがバイブル「マニュアル」の第4章に書いてある。

●まずは紙飛行機の形をデザインしてみるのだ

それでは，いよいよCAD攻略の第1段階として，まずは紙飛行機を作ってみよう。たった4つしか面がない簡単な物体にするつもりだから，あっというまにできるはずだ。

●たとえアトリビュートがひとつでも，アトリビュートの登録から始めるのだ

最初にアトリビュートを登録する。アトリビュートの色づけはATRで行うけど，面のアトリビュートの決定はこのCADで行うのだ。この点をよく理解しておいてほしい。

アトリビュートの登録には，まずAパネルのAttribute Modeのところをクリックしてアトリビュートモードに入る。次にその中のAttribute登録をクリックするとアトリビュート名の入力状態になるので，アトリビュート名を登録する。キーボードから「kami」と入力して(ち

ゃんとこのとおりに半角の小文字で入力すること)，リターンキーを押す。アトリビュートの入力が終了したら，ESCキーを押すとアトリビュートモードに戻る。

アトリビュート指定とは，これから新しく作る面のアトリビュート名を指定することである。最初の状態では，アトリビュート指定は atr no となっている。このままだと，作った面のアトリビュートが atr no になってしまう。そこで，アトリビュート指定を変更するために，アトリビュート名の登録が終わったら Attr. 指定変更のところをクリックして，アトリビュート指定を変更する。アトリビュートの指定変更には，マウスカーソルを「kami」と書いてあるところに持っていってクリックする。現在のアトリビュート名はCパネルのPanel 1の一番下に表示されているから，そこがちゃんと「kami」となっていれば，アトリビュートの指定変更は終了だ。

これでこれから作る面には「kami」というアトリビュートがつけられることになったわけだ。

●面を作るための基礎知識なのだ

面を作成する（物体を実際に作る）には，AパネルのEdit Modeをクリックして，エディットモードで行う。このCADで取扱う最小の単位は面で，物体は面を組み合わせることによって作っていく（つまり，できた物体は紙で作った張りぼてのようなものになる)。だからアンテナのような直線を作ることができないので，もしそんな物体を作ることになったら，とても細長い面を作ってごまかすしかない。

また，このCADには面の指定をするための面ポインタと，点を指定するための3Dカーソルがある。

現在ポイントされている面は，パネルBの「ポイント面確認」をクリックすると点線で表示される。新しく面を作るとポインタはその面に勝手に移動する。つまり，特に面のポインタを変更しなかったら一番最後に作った面に面のポインタがあることになる。

3Dカーソルは空間座標上の１点を示していて，その１点は各平面図の２つの白色の直線の交点で表されている。このCADでは面に対する操作はポイントされた面に，点に対する操作は3Dカーソルの座標に対して行われている。

物体を作るときはYZ平面が正面図となるので，この図面が物体を前から見た図となるように作成したほうがいい。別にほかの向きに作ってもかまわないけど，こうしておくと，フレームファイルを作るときにいちいち物体の向きで頭を悩ませる必要がなくなるので便利でいい。また，普通は物体の中心点が原点にくるように作るのがいいけど，家とか木などは底面の中心が原点になるように作成したほうが，フレームファイルが作りやすくなる。物体を作るときは物体の中心をどこに置くかもよく考えて，作成できるようになれば一人前だ。

●3Dカーソルの移動は，このCADの特徴なのだ

3Dカーソルの移動には，平面図にマウスカーソルを持っていってクリックすると，マウスの位置まで 3D カーソルが移動する。この移動の仕方がこの CAD の最大の特徴になっている。

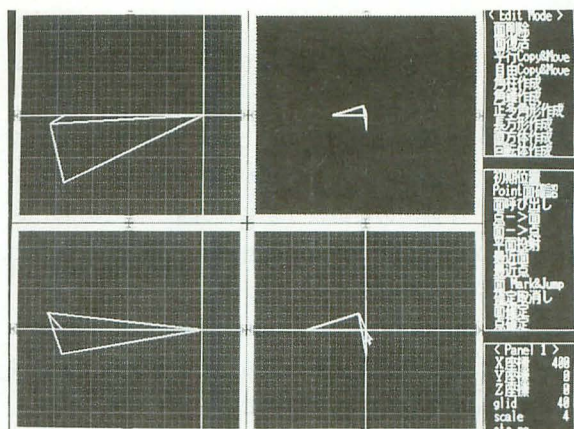



図2　3Dカーソルを使って飛行機の羽根を作っているところ。右上の透視図は，X，Y，Zの各軸まわりに回転させて向きを変えてみるとわかりやすい。

まず，マウスを移動したいところまで持っていって右クリックすると，3Dカーソルの縦の線がマウスカーソルの位置まで移動する。3Dカーソルは２つの直線の交点で表されているので，3Dカーソルの位置が横方向にだけ動くことになる。このときはマウスカーソルの縦方向の位置は関係ない。もちろんマウスの右ボタンを押したままマウスを左右に動かすと3Dカーソルの縦の線が一緒に動く（つまり3Dカーソルの位置が左右に動くことになるわけだ)。

3Dカーソルを縦方向にだけ移動したいときは，マウスを左クリックする。つまり右クリックと左クリックをすれば3Dカーソルはマウスカーソルの位置に移動するわけだ。これだとマウスのボタンを２回クリックする必要があるのでめんどくさいかもしれないが，この移動の方法は，たとえば3Dカーソルを縦や横にだけ移動したいときなどに非常に便利だ。

また，ある点に 3D カーソルを移動したい場合，まず3D カーソルの縦の位置だけを合わせて（つまりマウスの左ボタンだけをクリックして)，そのあと横の位置を合わせて思った点に移動する，ということができる。マウスの右ボタンと左ボタンを両方押したままマウスカーソルを動かすと3Dカーソルも一緒に動くけど，できればそんなことはしないで，早くこの CAD の3Dカーソルの移動方法に慣れてほしい。

●いよいよ面を作るのだ

面を入力するには２つの方法がある。面の頂点をひとつずつ決定していく方法と，Edit Modeのコマンドを利用する方法だ。やはり初心者はコマンドを使うなんて横着はせずに，頂点をひとつずつ入力していくほうがいい。この方法さえできるようになれば，根性次第でどんな物体でも作れるようになるのだ。何事も基本が第一である。

まずは紙飛行機の羽根を作ろう。AパネルのEdit Modeをクリックしてエディットモードに入る。このモード以外では面は作れない。

エディットモードに入ったら，マウスカーソルを平面図の真ん中より右のほうに持っていってそこで左クリックする。すると3Dカーソルが右に移動する。Cパネルに現在の3Dカーソルの座標が表示されているので，それを参考にして，3Dカーソルの座標が（400，0，0）となるようにする。平面図の右のほうがX軸の正方向なので，それよりもX座標が小さい場合は3Dカーソルをもっと右に，大きい場合は左に移動させればいい。

また，写真では見えずらいが平面図，側面図，正面図にはバックに青い線でマス目がある。CパネルのPanel 1にあるglidがそれだ。これは40ごとに引いてあるので，X軸の正方向へ10マスでちょうど（400，0，0）になる。

3Dカーソルが（400，0，0）に移動できたら，そこでスペースキーを押すとAパネルに表示が出て面入力状態になる。スペースキーとBパネルの点確定は同じ機能だけど，いちいちマウスを動かさないでいい分スペースキーのほうがお得だ。

こうして面入力状態に入ると，現在の3Dカーソルの位置が面のひとつ目の頂点ということになる。三角形を作るのだからあと2つの頂点を決定しないといけない。そこで3Dカーソルを（−360，−320，0）に移動させる。今回は，左クリックだけじゃなくて右クリックを使って縦方向にも移動させないと（−360，−320，0）に移動できない。さっき確定した点から黄色い点線が現在の3Dカーソルの位置まで延びているけど，これが作っている三角形の辺になるのだ。

3Dカーソルを（−360，−320，0）に移動させたら，再びここでスペースキーを押して点を確定する。すると黄色い点線が黄色い実線になる。こうなって初めて，2つ目の点の確定が終わったことになる。3つ目の点は（−440，−40，80）なので，いくら平面図でがんばっても，Z座標は80にならない。そこで，マウスカーソルを正面図の真ん中より2マス上のところに持っていって右クリックする。Z座標が80になったので，あとは再び平面図に戻り，3Dカーソルの位置を（−440，−40，80）に持っていってスペースキーを押すと3つ目の点の位置の確定が終わる。3角形を作るのだから，点の決定が3つ終わったらリターンキーを押し面を確定する。すると今作った3角形が緑色で表示される。

我々はついに面の作成に成功したのだ！

あとは，3つの頂点がそれぞれ，

（−440，−40，80），（−360，0，−120），（400，0，0）となる3角形を作ったら紙飛行機の左半分が完成する。右半分もY座標の正と負を逆にした面をそれぞれ作れば，それで紙飛行機の完成だ。

●物体作成の終了

物体の作成が終了したらEdit Modeをクリックしてモード選択に戻り，今度はQuitを選ぶ。そしてSave & Exitのところで左クリックすると，セーブするファイル名を聞いてくるので適当にファイル名を入力してリターンキーを押す。これでいままで作ったデータがセーブされてCADが終了する。あとはAUTOで自分の作った物体を見てみるのもいい。

●これで終わったと思うなよ

う〜む時間がない。残念ながら今回のCAD攻略はここまでである。しかし，今回ここで学んだことを利用すれば，CADの完全攻略は目の前である。

それでは，勇者諸君の健闘に期待する。

CADテクニック編

●いかにしてCADを使いこなすか

正直にいってCADを使いこなすにはある程度センスが必要です。2，3日で複雑な物体を自由自在にデザインできるようになる人もいれば，なかなか進歩しない人もいます。しかし要は「慣れ」の問題なのですから，1時間でも多くCADに接することが大切です。

CADには非常にたくさんの機能がありますが，それらの機能をひとつでも多く知っていることが使いこなすコツ……ではありません。むしろ，いくつかの大切な機能を徹底的に使い込むほうが実戦的です。マニュアルではすべての機能をズラズラッと解説しているので，どの機

あき姫の迷える子羊のコーナー

はい，こんにちは。姫でございます。Q&A，行きま〜す!!（アムロの口調で）

小羊：Please send 'DōGA CGAsystem' to Canada for me？（和訳：DōGA CGAシステムをカナダまで送ってくれませんか？）

（カナダのMr. S）

姫：U〜Mu. Ma iidesu kedo. koukubin ha takaina……. hunabin jya dame desuka? kono sai, eigoban no manyuaru mo tukutta houga iinde naikai.（和訳：う〜む。まあいいですけど。航空便は高いなあ……。船便じゃだめですか？この際，英語版のマニュアルも作ったほうが……）

小羊：消費税はどうなるんでしょう。まさか＋ってことはないでしょうね（まったく，貧乏学生から税金なんかとるなよな）。（N.T.）

姫：来ると思った，この質問。そんなわけはないでしょ……と言いたいところだけど，マニュアルの印刷や発送の業者には，しっかり取られたはずだな〜。まあ，その分は実費の計算に入っているので，安心（？）してください。

小羊：「あき姫」の名前の由来は何ですか？

（某編集長　M）

姫：本名が「あきこ」で，昔のペンネームが「ひめ」だったからです（単純明快）。

小羊：あの〜，カンパとして，ビール券と図書券を送ってきた人がいるんですが，どうしたらよいでしょう？

（某プロジェクトチームのスタッフ）

姫：姫の担当でないのでわかりかねます。会計課のほうへおたずねください。個人的に言わせてもらえば，姫はビールより水割りのほうが好きです。

（会計課より：当方では，物々交換はお取扱いしておりません）

というわけで，今月はこれでおしまい。お手紙お待ちしています。

能が大切なのかよくわからないでしょう。「回転体」などは一見便利そうですが，使える機会が限られ，面数が増えるばかりであまり実用的ではありません。

特に重要なものを挙げると，以下のようになります。

Aパネル	面削除	Attribute登録
	object 初期化	
Bパネル	Point面確認	面呼び出し
	平面投射	最近点
	面確定	点確定
	確定取り消し	
Cパネル	scale	m. cont.
	m. cont. ＊2	pen color

また，次に重要なものを挙げると以下のようになります。

Aパネル	角柱作成	面のattr.変更
	attr.の色づけ	
Cパネル	c. speed	c. speed ＊2
	マクロ（パネル５）	

これだけ知っていれば通常の制作には不自由しないはずです。その他の機能については，どんなものがあるかを知っておいて必要なときにマニュアルを調べれば十分でしょう。

●CADの禁じ手

1)　穴のあいた面

壁に窓をつけるときなどは，穴のあいた面が必要になります。しかしこのCADではそのような面はサポートしていません。そこで，壁を複数の面に分割することで解決します。

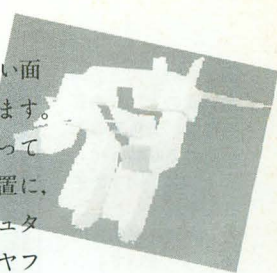
2)　同一平面の含まれる２つの面

ある物体の表面にマークなど入れるために，ある面に重ねて別の面を作る人がいますが，これでは作画時に問題が発生します。たとえば次ページ図3のように，四角い面（XY平面：Z＝0）の上に三角のマークを入れるとします。このとき，マークの三角形もXY平面（Z＝0）で作ってはいけません。そんなことをすると，「空間のある位置に，同時に複数の物体は存在しない」と言ったアインシュタインの立場がなくなってしまいます。CADではワイヤフレームしか作成しませんが，物体の面に色をつけた場合，作画プログラムは１ラインごとにどちらの色で作画すればよいのか迷ってストライプが発生します。

このような場合，２通りの解決方法があります。まずひとつは，三角形を四角の面から平行に少し浮かせておく方法です。同一平面でなければよいのだから，浮いているのはほんのちょっとでよいはずですが，実際には誤差の問題がありますので，全体が1000ぐらいの物体なら10くらい離してみてください（マス目の設定を変えると差はもっと微妙にできます)。この方法では，横から眺めると，三角形が浮いているのが見えたりするのですが，実際にやってみるとほとんど気になりません。

もうひとつは，四角の面に三角形の穴をあけ，その穴に三角形をはめ込む方法です。もちろん穴のあいた面はできませんので，実際には分割することになります。しかし非常に面倒な作業となりますので，あまり現実的ではありません。

3)　交差する２つの面

これは絶対にしてはいけないというほどのものではありません。しかし交差しているところは，複数の面が存在することになり，その１ドットの色がどちらの色になるか，さだかではありません。その結果，境界線がギザギザになってしまう可能性があります。これを避けるには，どちらかの面を分割すればよいのです。

ただいま，会員募集中

大阪工業大学グラフィック研究会

さて，今回紹介いたしますのは，プロジェクトチームDōGAと淀川をはさんでのお向いさんにあたる「大阪工業大学グラフィック研究会」(略称：GR）です。なにしろ自転車で30分ほどのところにあるので，当チームとの交流もたいへん盛んに行われています。代表の森山さんは，学会などにもいくつか論文を発表しているぐらいハイレベル＆アカデミックな方です。ではどうぞ。

はじめまして，私たちはグラフィック研究会です。

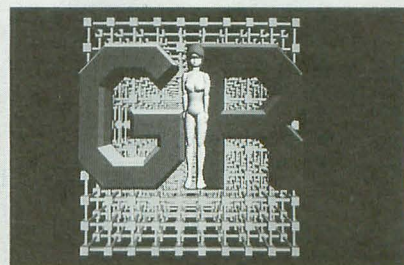
当会は大阪工業大学の学生有志による CG の研究団体で，1982年発足という大学の CG 研究サークルとしては比較的古い時期に研究活動を開始して以来，７年間にわたって現在もなお活動を行っております。

これまでの主な活動としては，GDC（μPD7220）を利用したリアルタイムCGアニメーションシステムの開発を行ったり，ソフト面では16,000面という面データによる人体モデルとその歩行シミュレーション映像を作成して専門誌に紹介されました。

そして現在，新しいアイデアのCGAシステム「ANGEL(エンジェル)」の開発をX68000上で行っております。このシステムは以下の特徴を持っています。

BASICライクなプログラムによるCGアニメの記述・実行が可能です。

独自開発の「Sバッファ法」によって，22,800ポリゴンをアンチエリアシング処理付きで３時間30分で描画できます（標準システム＋ハードディスク構成で可能→RAMディスクならさらに高速実行)。

当会ではこのようなシステム開発の他に，各種CGシステムを使ったCGアニメーション作品の制作や CG の建築シミュレーションへの応用といったテーマを各メンバーが自由に行っています。また，1987年からDōGAプロジェクトにも

参加しており，さらに同年コンピュータグラフィックス連合にも参入，当会はその大阪支部として活躍しています。

これからも CG の研究やアニメーション制作活動を通じ，いろいろな CG サークルやアマチュア CG 作家の方たちと知り合いたいと思いますすので，どうぞよろしくお願いします。

また，大阪工業大学の学生さんで興味のある方はぜひご連絡ください！

連絡先：〒535 大阪市旭区中宮5-15-20　曙荘
グラフィック研究会　代表
森山弘樹

図3　XY平面上の四角い面の上に三角形を作る。三角形の3頂点のZ座標は10。
図4（右）　図3の正面図を拡大したところ。三角形が四角の面から「浮いて」いるのがわかる。

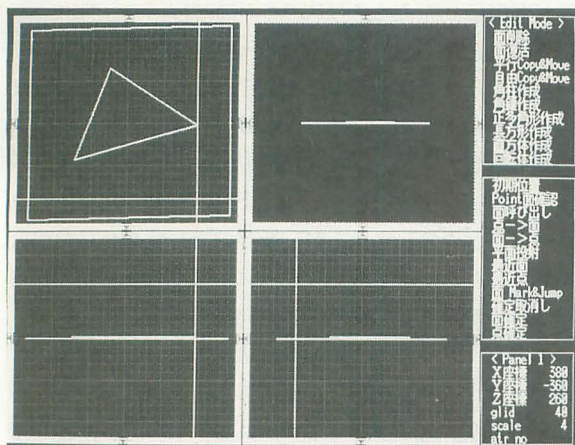
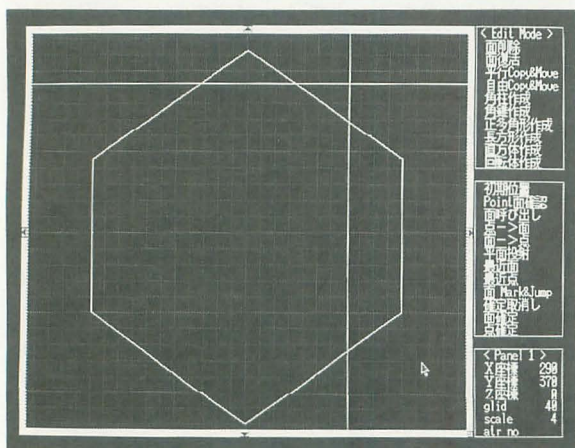



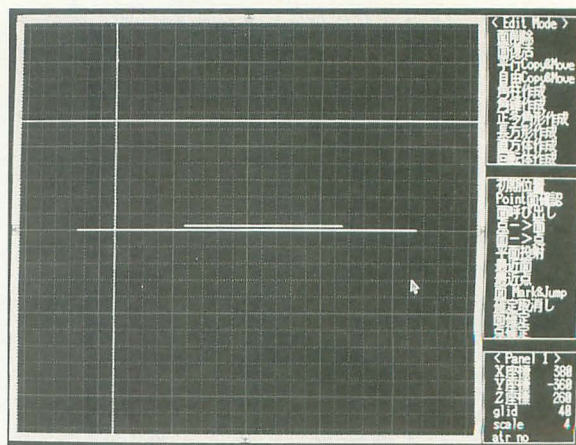
図5　バックのグリッドを使いだいたいの見当で作成した（正）六角形。

●作業効率の向上

1）　キーボードの活用

一般にキーボードよりマウスのほうが操作性がよいと思われていますが，実際にはキーボードのほうがよいケースもたくさんあります。このCADにおいては，すべての機能がキーボードで実行できるようになっていますので，マウス（右手）とキーボード（左手）をうまく使い分けることが，操作性を向上させるコツといえます。

以下の機能については，常にキーボードで操作するように心がけてください。

面確定：	リターン
点確定：	スペース
点確定取り消し：	BS
Point面確認：	「2」（テンキーでないほう）
面呼び出し：	「3」（　〃　）
面呼び出し（逆）：	SHIFT＋「3」（　〃　）
平面投射：	「6」（　〃　）
最近点：	「8」（　〃　）
面削除：	DEL
操作図面の変更：	「A」（平面図），「Z」（側面図），「X」（正面図），「S」（透視図）
透視図の物体の移動，回転：	テンキー，四則演算キー

ご覧のように，前述の「特に重要な機能」の大部分がキーボードで操作することになり，マウスはほとんど3Dカーソルの移動に専念できます。

2）メッシュの活用

メッシュとは3面図に表示されている青のマス目のことです。起動時には，1マス（glid）が40に設定されていますが，この値では100とか300という座標がちゃんとマス目上にこないので私は嫌いです。そこでCパネルのPanel 1のscaleを右クリックしてください。1マスが20になります。また，3Dカーソルは起動時に10ずつしか動かないようになっています。

初心者は小さな凹凸や細部にこだわりすぎる傾向があります。どんなに細かいところに凝っても，最終的にアニメーションになったときにはほとんど変わりません。もっとデフォルメして，抽象化して，単純化してください。形状デザインの最大のポイントは，複雑な形状をいかにそれらしく単純化するかということです。

モデリングするときは，物体のすべての頂点が，きちっとメッシュのマス目の上に来るように心がけてください。それができないような面などは，思い切って省略するぐらいのつもりでよいと思います。それによって，操作性は飛躍的に向上します。

3）　正多角形の作り方

よく使う正多角形は，三，四，六，八角形です。四角形つまり正方形は誰にでもできるので話を別にするとして，三，六，八角形はどうやって作ればよいでしょうか。

CADには「正多角形作成」という機能があるのでこれを使えば簡単だ……なんて思っているようではまだまだです。「正多角形作成」では厳密に正しい図形を作成しますので，それらの頂点の座標はけっしてメッシュのマス目の上にのってくれません。正多角形を作るのは簡単でも，その後の作業効率が著しく低下してしまいます。

モデリングにおいて必要なことは，厳密に正しい図形ではありません。それっぽく見えさえすればよいのです。図5はマス目を使って適当に作成した六角形の平面図ですが，よく見ると明らかに厳密な正六角形ではありません。でも十分それらしく見えると思います。

●同一平面の問題

このCADはサーフェスモデリングツール，つまり面を

基本単位として物体を構築していきます。各面は何角形でもよいのですが，面である以上，各頂点は同一平面上にないといけません。しかしながら，ひとつの面のすべての頂点を同一平面上にのせることは意外と面倒なのです。適当に面を作っている途中で点確定しようとすると「ピッ」と文句をいわれることがあります。これはその点がちゃんと同一平面にのっていない証拠です。

もちろん，XY，YZ，ZX 平面のどれかに平行な面ならば入力は非常に簡単です。しかし，どの平面にも傾いた面の場合はどうしたらよいでしょうか。

1) 斜めの面は使わない

とはいえ，まったく使わないという訳にはいきません。安易な気持ちで使ってはいけないということです。そして，斜めの面を使う場合は必ず簡単な比の傾きにしておくということが大切です。

傾きの比が簡単というのは，たとえば，「XY平面には垂直で，X方向に2進んだらZ方向に1進む」とか,「どの平面に対しても垂直でないが，X方向に1進むとZ方向に－1またはY方向に1進む」などという言い方で傾きを表現できるということです。

このようにしておくと，難しい計算なしに簡単に同一平面にのせることができ，しかも各頂点が，メッシュのマス目の上に合うようになります。

2) 三角形に分割する

ピタゴラスさんの言うことを信じると,「一直線上にない3点はひとつの平面を構成する」そうです。つまり，三角形なら必ずすべての頂点が同一平面上になるのです(あたりまえか)。

ですからうまく同一平面にのらないようならば，全部三角形に分割してしまえばよいのです。ただ，このとき分割の仕方で形状が若干変わりますので注意が必要です。

3) 平行線を含む四角形

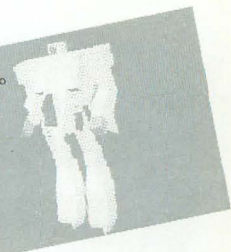
ピタゴラスさんがくれたもうひとつのアドバイスに,「一対の平行線はひとつの平面を構成する」というのがあります。つまり一対の平行線のある四角形（簡単にいうと台形）の各頂点はかならず同一平面上にあるということです。これを利用すると，四角形の各頂点を簡単に同一平面上に置くことができます。

4) 平面投射

平面投射は，まさに同一平面の問題を解決するために設けられた機能です。うまく同一平面にのらないときには，「6」のキー（テンキーでないほう）を押してください。3Dカーソルが同一平面になるような位置に，自動的にずれてくれます。少し注意が必要なことは，このずれ方がそのときの操作図面によって異なるということです。

操作図面というのは，4面図（3面図＋透視図）の内のひとつで，そのときのペンカラーの点線で囲んで表示されているものです（点線はよく注意して見ないとわかりません)。

ずれ方は操作図面（ただし透視図は除く）に対して垂直方向になるので，たとえば平面図が操作図面の場合，3DカーソルはZ軸（高さ）方向にのみ動いて同一平面上になります。ですから，上から見てこの位置に頂点を作りたかった場合，平面図を操作図面にしてから平面投射を行ってください。

一見非常に便利なのですが，この機能はあまり使わないほうがよいでしょう。簡単に同一平面にできるため，安易な気持ちで斜めの面を作ってしまいます。平面投射

INFORMATION

なぜCGAシステムが届かないのか

パターン1　まだ申し込んでいない

届かないのはと～ぜんです。申し込みを希望される方はお早めに。一応，申し込みは10月末までということになっておりますが，マニュアルの増刷などは数が揃わないとできないので，なくなったところで終了してしまうかも知れませんよ。

パターン2　申し込むのが遅かった

どのくらいの応募があるかよくわからなかったので，8月1日発送分は少な目に用意していました。それが予想外の反響だったため，7月になって本当にすぐ申し込んだ方にしか送れなかったのです。あわてて追加の準備をし，1，2週間遅れて2回目の発送を行う準備をしましたが，それでも7月中に応募された方のすべてには到底対応できません。ただ今，3回目の発送を準備中ですので，7月後半以降に申し込まれた方はいましばらくお待ちください。

パターン3　申し込みが届いていない

自分では申し込んだつもりでも，当チームに届いていない場合があります。書留や郵便振替なら大丈夫ですが，普通郵便で申し込んでいるなら，途中でなくなることもあるかもしれません。思い当たる方は，往復葉書で，当チームの「アフターサービス係」まで，問い合わせてください。

パターン4　送り先を書くのを忘れた

私も郵便振替なんて知らなかったのであまり人の事は言えませんが，振替用紙のご自分の住所と名前を書く欄に，当チームの住所と私の名前を書いて，どこにも発送先を書いていない方が数名いらっしゃいます。あのう，私はどこに送ればよいのでしょうか？　思い当たる方は，振り込んだ郵便局名と日付を明記し，筆跡鑑定用に(？)「鎌田優」と数回書いて当チームまで送ってください。

パターン5　カンパをしなかった

申し込みは，実費3000円＋カンパ（1口1000円：1口以上）ということになっています。それなのに実費3000円しか振り込んでいない方が2，3名いらっしゃいます。カンパなんて強制するものではないし，カンパを目当てに活動しているわけではないので別にかまわないという見方もできますが，申し込みの条件をよく思い出してください。このシステムや当方の活動がカンパに値しない，つまり賛同できないという方のご応募はご遠慮願っています。

カンパがなかったのは，たぶん何かの勘違いだと思うのですが，とりあえず発送は保留になっていますので，ご連絡ください。

パターン6　発送にトラブルがあった

発送は宅配便などを使用しています。届いたかどうかのチェックは発送業者のほうで確認していますので，まず問題はないでしょう。ずっと不在にしていたりすると，届きようがないかも知れません。

パターン7　DōGAに非常事態が発生した

このパターンだけは避けてほしい。今のところ特に大きなトラブルはなく，予定通りに進んでいます。しかし今後，落雷によってすべてのプログラムが消滅するとか，プロジェクトルームがゴジラに襲われるとかいうこともあるかもしれません。そのような場合，DōGAに問い合わせてもまったく音信不通になっているでしょう。そのようなときは，ただひたすら神に祈りを捧げましょう。

によってできた頂点の座標は，メッシュのマス目の上に来ているわけがありません。平面投射は，使えば使うほどマス目の上にのらない点が増え，操作性が低下し，さらに斜めの面が増え，また平面投射を使わなければいけなくなるという麻薬のような性質があるのです。この機能はあくまで最後の手段として用いてください。

来月はアトリ+αだーっ

さて，今回はCADを使いこなすテクニック編などをお届けしました。だいぶ本格的なCGAらしくなってきたように思うのですが，CADを相手に悪戦苦闘している皆さんのお役に立てたでしょうか。上手く使いこなせなかった方も，もう一度チャレンジしてください。CADのテクニックなどは，この程度で尽きるようなものではありません。皆さんも自分なりのテクニックをどんどん開拓していってください。

来月はアトリビュートデザインのテクニックを中心に解説します。アトリビュートデザインは簡単なようで難しい，なかなか「通」なところです。ATR（アトリビュートエディタ）の使い方は完全にマスターしているものとして話を進めますので，ちゃんと練習しておいてください。それからおまけとしてCGAシステムの隠し機能をいくつか紹介しましょう。お楽しみに。

秋の学祭用にCGA制作に取り組んでいる方も多いと思います。順調に進んでいますか？　やはりいきなり大作に挑むのは無謀というものです。とりあえず今年はできたカットをつなげて，客よせのデモを作るぐらいの気持ちで取り組んだほうがよいでしょう。面白いものができたら，ぜひ拝見させてください。お便りお待ちしています。

わ〜い，わ〜い，にこにこ　ぷん！

★DōGA・CGA システムは一般のお店では取り扱っていません。私達の活動に賛同してくださるアマチュアの方には，カンパ（1口：1000円より）と実費（3000円）だけで配布しています（プログラムは無料です）。郵便振替で申し込んでください。

申し込み期間：1989年7月1日〜10月30日
郵便振替口座：大阪 3-109598　口座名：鎌田　優
または，DōGAプロジェクトルームCGA システム配布係に直接現金書留で申し込んでいただいても結構です。
なお，発送は申し込まれた順番に行いますので，場合によっては少し遅れることがあります。

★CGA システム，本連載，各コラムについてのお手紙お待ちしています。

〒533 大阪市東淀川区淡路5-17-24 篤コーポ102号室
DōGAプロジェクトルーム「なんでもどんとこい」係

●CGA教養講座　その1「世界の最先端のCGA作品の動向」●

世界の最先端のCGといえばSIGGRAPH（シグラフ）である。SIGGRAPHはもともとは純粋なるCGの学会であり，最新のアルゴリズムやハードを発表する場であった。しかし，そこのフィルムショーで最新の映像が公開されるとなると，そこに人気が集中し，いろんな人が見に来るようになり，ほとんど単なるお祭りになってしまったのだ（アメリカ人はなんでもお祭りにしてしまう）。

今年でまだ10年ぐらいのものだが，幸いにしてその間のほとんど大部分のフィルムを見ることができた(もちろん国内で)。そこで最新の作品を中心にしながら，最先端のCGAの動向をまとめてみた。以下の内容については私の独断といわれる方もいらっしゃるだろうが，少なくとも私はこう感じたのだ（だからそれが「独断」なんだってば)。

1)　レンダリングアルゴリズムの追求

まだ地球上で巨大な恐竜たちが全盛を誇っていたころ，レイトレースのアルゴリズムが発表され，チェッカーボードの上を飛び跳ねる透明の球が流行したものだ。当時，グラフィック・ワークステーションといえども今のパソコンに毛がはえたようなもので，CGはなかなか手に余っていた。だからSIGGRAPH においても，いかにアルゴリズムを工夫して（手を抜いて）それらしい絵を高速に表示するかというテーマが注目を浴びた。そこで，さまざまなスムースシェイディングやマッピングが次々と発表されていた。レイトレースにおいても高速化は論議の的で，空間分割法（等分割，オクトリー）などが発表されたのである。

現在の市販のパソコンCGソフトはちょうどこのころの技術を実用化しているといえるだろう。C-TRACEはバンプマッピングを，サイクロンは等分割法をそれぞれ使用することができる。

しかし，この流れは次第に下火になってしまった。なぜなら，ワークステーションの能力が向上し，あまり計算スピードを気にする必要がなくなってきたからだ。それよりもいかに現実に忠実な画像を生成するかというところに興味が移っていった（初心者の中には，レイトレースは完璧で完全に現実と同じ画像が生成できると思っている人がいるようですが，とんでもない間違いだ)。その代表がラディオシティー法で，物体間の相互反射（照り返し）などを忠実に再現した。これは主に建築の分野で，部屋の環境シミュレーションなどに利用されている。

ここまで書いて気がついたのだが，ぜんぜんCGA作品の解説をしていない。そこで，現実に忠実な描写を描いて注目された作品のひとつ「FOG CITY」を紹介しよう。これは日本が誇るCG界の3教授のひとり，広島大学の中前教授の作品で，文字通り霧の中の街を表現している。日が暮れるにつれて霧が出，すべての物がぼやけていく。通り過ぎる車のヘッドライトがかっこいい(特にストーリーはない)。

ただ残念なのは，ここ数年こればっかりやっているため，ついに去年はSIGGRAPHフィルムショーに参加できなくなってしまった。しかし中前教授は大変偉い方なので，私ごときが「もっと変わりばえのあるのを作って」などと偉そうなことを書いたりは決してできない。

しかしながら，この流れもすでに主流ではない。レンダリングアルゴリズムの追求自体がもう下火になってしまったのだ。

（次回に続く）

X1用シューティングゲーム

Defeat X

Asano Hidefumi
浅野 英史

流れる星をバックに敵巨大戦艦に挑む横スクロールタイプのシューティングゲームです。豊富なオプション兵器を使いこなして敵機を撃ち落してください。CTCを持ったマシンではBGM対応にもなります。

シューティングだ

X1ユーザーの方お待たせしました。

「Defeat X」はX1用の横スクロール型シューティングゲームです。全4面で，自機の装備をその面の特性に合わせて選択しながら，最後のデカキャラを倒していくという，シューティングゲームとしてはオーソドックスな構成ながら，なかなかゲーマーの血を熱くしてくれるゲームです。非パワーアップ型グラディウスといった雰囲気だと思ってもらえればいいでしょう。

4面構成，BGM，ボスキャラつきでもプログラムは6Kバイト程度ですので，打ち込んでもなくならないダンプリストに苦しむこともないでしょう。ぜひ打ち込んで遊んでみてください。

入力方法

まず，最初にひと言。このプログラムはすべてCZ-8FB01（X1用ディスクBASIC）V1.0で動作するようにできていますから，CZ-8FB01以外では使用しないでください。

では入力方法です。リスト1からリスト6までのうち，リスト1，リスト2，リスト4はBASICプログラムですから，CZ-8FB01を立ち上げて，そのまま入力してセーブしてください。リスト3，リスト5，リスト6はマシン語ですから，マシン語モニタまたはMACINTO-Cなどのマシン語入力ツールを使って入力していきます。チェックサム，CRCチェックバイトをよく確認したうえで，これらのプログラムはそれぞれ同じ1枚のディスク上にセーブしておいてください。

また，SAVEする際，リスト2からリスト6まではファイル名が決まっていますから，必ず，

リスト2：SAVE"MUSIC DRIVER.Bas"

リスト3：SAVE"MUSIC DRIVER.Bin",&HF800,&HFA66

リスト4：SAVE"SP SUB.Bas"

リスト5：SAVE"SP MAIN.Bin",&HD000,&HDE37

リスト6：SAVE"ROUND DATA.Bin",&HE500,&HE8FF

のようにしてセーブするようにします。

リスト1を実行すれば，自動的にファイルを呼び出し，しばらくするとゲームが始まります。

操作法

さして難しい操作は必要ありませんが，プレイには2トリガーのジョイスティックが必要です。ふつうのシューティングゲームのように，ジョイスティックで自機を上下左右に動かして（8方向です），Aボタンで弾を撃ちます。GAME画面の右上，残機数の左横には，現在自分が選択している装備が示されています。

装備にはNORMAL，TWIN，UP-DOWN，3WAY，BEHINDの5種類があって，それぞれ弾の出る方向と数が違います。各装備の機能は図1のようになっています。画面上には同時に6発の弾しか出ませんから，敵キャラや状況によってこれらを使い分けていくことが攻略のポイントです。このあたりは実際にやってみたほうがよくわかるでしょう。

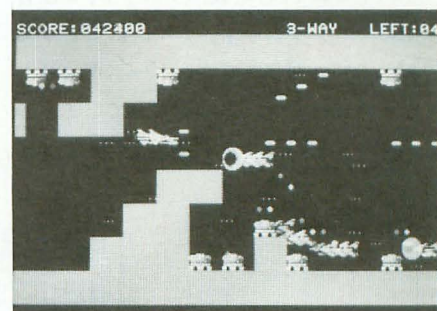
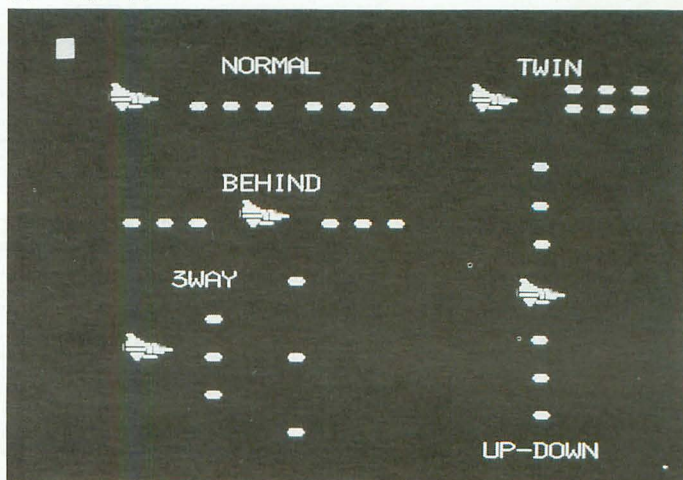



図1 武器一覧

この装備はゲームの最初からすべて使用でき，選択はジョイスティックのBボタンでいつでも変更可能です。

それから，タイトル画面が出ているときは，ジョイスティックの上下でゲームのスピードを調節できますし，ジョイスティックの左右で面の選択ができます。スピードは，9（かなり遅い）から1（超人的な速さ）までお好みのままどうぞ。面は1から4面まで取り揃えております。

各面は強制スクロールによる敵基地攻略

と敵機迎撃，ボスキャラで構成されています。各面の最後には，大型のボスキャラが威厳をもってあなたを迎えてくれますから，中心部に弾を打ち込んで破壊してあげてください。これでその面がクリアできます。すべての面をクリアすると感動のエンディングがあなたを待っています。

プログラムについて

リスト1はPCG定義，リスト2，3は音楽用のプログラムです。リスト4，5が本命のゲームプログラムです。ゲーム中のほとんどの処理はマシン語で行われていますので，プレイ中のブレイクキーは効きません。

プログラムのうち，BASICでやっているのは，タイトル部分とエンディングぐらいで，そのほかはすべてマシン語で動作しています。

なお，このゲームはCTCをつけている機種ではPSGによるBGMつきになります。FM音源ボードやX1turbo内蔵，立体視ボードなど，それぞれでCTCのアドレスが違いますから，リスト2の150行を注釈を見て自分の機種用のアドレスデータに変更してください。デフォルトはX1のFM音源ボードになっていますので，たとえば，X1turboであれば，150行の4，7を160，31に変えるわけです。

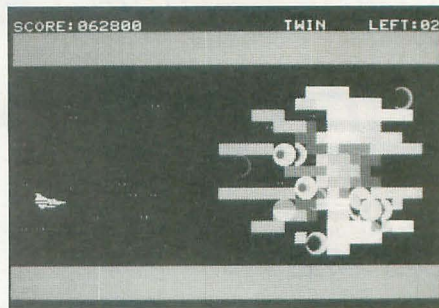
それでは，X1では久々のシューティングゲームをじっくりと楽しんでみてください。

Profile
◇浅野さんは大阪府にお住まいの18歳，高校3年生です。マイコン歴は約5年，MZ-700を経て，現在X1turboIIのユーザーです。今度は縦スクロールにも挑戦したいとのこと。

リスト1 SP INIT.Bas

```
10 WIDTH40:CLS:CSIZE1:CFLASH1:LOCATE10,10:PRINT#0"Wait a moment,please."
20 FOR i=32 TO 90:DEFCHR$(i)=STRING$(3,LEFT$(CGPAT$(i),8)):NEXT
30 A$(0)=STRING$(8,CHR$(0)):PALET 1,0:OPTIONSCREEN 2:
40 A$(1)="UxUxUxUx":A$(2)="xUxUxUxU":A$(3)="〒〒〒〒〒〒〒〒"
50 READ A$,B$:A$=A$+B$:FOR I=0 TO 34:B$=BIN$(VAL("&H"+MID$(A$,I*2+1,2)))
60 FOR J=1 TO 3:A(J)=VAL("&B"+MID$(B$,J*2+1,2)):NEXT
70 DEFCHR$(177+I)=A$(A(1))+A$(A(2))+A$(A(3)):NEXT:B$=A$(0)
80 DEFCHR$(33)=HEXCHR$(STRING$(3,"0000007EFF7E0000"))
90 DEFCHR$(34)="〒〒〒〒〒〒〒〒"+HEXCHR$("FF818D85818181FF")+"〒〒〒〒〒〒〒〒"
100 DEFCHR$(35)=B$+HEXCHR$(STRING$(2,"0000183C3C180000"))
110 FOR I=97 TO 120:READ A$:DEFCHR$(I)=HEXCHR$(A$):NEXT
120 C$=STRING$(16,"0"):FOR I=121 TO 146 STEP 2:READ A$:GOSUB200:DEFCHR$(I)=HEXCH
R$(RIGHT$(C$+A$+A$,48)):DEFCHR$(I+1)=HEXCHR$(RIGHT$(C$+B$+B$,48)):NEXT
130 FOR I=0 TO 7
140 A$="00000000"+RIGHT$("0"+HEX$(2^I),2)+"000000":GOSUB 170:NEXT
150 RUN "MUSIC DRIVER .Bas"
160 '------------------------------------------------------------ SET STAR PAT.
170 B$=STRING$(16,"0"):DEFCHR$(I)=HEXCHR$(A$+B$+B$)
180 DEFCHR$(I+8)=HEXCHR$(A$+B$+A$):DEFCHR$(I+16)=HEXCHR$(A$+A$+A$):RETURN
190 '------------------------------------------------------------ SET REVERSE
200 A=1:B$="":FOR J=0 TO LEN(A$)/16-1:FOR K=7 TO 0 STEP -1
210 B$=B$+MID$(A$,J*16+K*2+A,2):NEXT:A=A+0:NEXT:RETURN
220 '------------------------------------------------------------ PCG DATA
230 DATA"B08CBC83B38FBF9084948191859598B892B2"
240 DATA"96B6AC86A68EAE96B69EBEA38BAB9BBBAF"
250 DATA"000000381C1F3F3F 000000381C1F3FBF 000000381C1F3F3F"
260 DATA"000000000000C0FB 000000000000C0FB 000000000000C0FB"
270 DATA"0000000000000000 00000000000000C0 0000000000000000"
280 DATA"003F000F07030000 80BF000F07030000 003F000F07030000"
290 DATA"1DED0DC0BF000000 1DED0DC0BF000000 1DED0DC0BF000000"
300 DATA"80FFF00000000000 F8FFF00080000000 80FFF00000000000"
310 DATA"00000000000001FF 00000000000000F8 00000000000001FF"
320 DATA"000000000000E3F1 000000000000677F 000000000000E3F1"
330 DATA"0000060E1870FEE0 0000060E1E7EFEFE 0000060E1870FEE0"
340 DATA"1F00000000000000 1F01000000000000 1F00000000000000"
350 DATA"F06F01003F000000 F0EF01000F000000 F06F01003F000000"
360 DATA"00FCFE3FCF000000 0155EA35CE000000 00FCFE3FCF000000"
370 DATA"000000000010FFF 0000000000000000 000000000010F00"
380 DATA"0000000000E3F0EF 0000000000040000 000000000084C008"
390 DATA"0000020E7CFC00FC 0000041000000203 0000000000000000"
400 DATA"0300000000000000 0000000300000000 0000000000000000"
410 DATA"F70300FF00000000 0000000000000000 0403000000000000"
420 DATA"FCFCFC3B00000000 0200000000000000 0000C03800000000"
430 DATA"000000000001061D 0000000000010001 000000000001061D"
440 DATA"000000038E0DB80 000000038E0DB80 000000038E0DB80"
450 DATA"0000000E30FCFC18 0000000E3EFCFC18 0000000E30FCFC18"
460 DATA"7F07000100000000 7F1F000000000000 7F07000100000000"
470 DATA"3F7F1FE7F9000000 1F1F070100000000 3F7F1FE7F9000000"
480 DATA"800080ECF4700000 C0F0F0E070000000 800080ECF4700000"
490 DATA"071F3F7F7FFFFFFF","E0F8FCFEFEFFFFFF","071F3F7F7FE1C0C0"
500 DATA"071F3F7960C0C080","E0F8FCFE7E3F3F1F","071F304040800000"
510 DATA"E0F8FC3E1E1F0F0F","0718000000000000","E0781C0E06070303"
520 DATA"00000000071E3E7E00000000061C18100000000075FBBFF"
530 DATA"0000000E0787C7E00000000008683810000000040A04820"
540 DATA"7F7F007F77E3E3000000007D04E3C2E3757A007F14E3E300"
550 DATA"FEFE00FEEEC7C70000000040004080C7409000FE28C7C700"
```

リスト2 MUSIC DRIVER.Bas

```
10 CLEAR&HF7FF:LOADM "MUSIC DRIVER .Bin"
20 '-------------------------------- TONE READ
30 PLAY 7200:DIM A$(11):FOR I=0 TO 11:READ A$(I):NEXT
40 PLAY "V0":ADR=&HFA88
50 FOR I=1 TO 8:PLAY "O"+RIGHT$(STR$(I),1):FOR J=0 TO 11:PLAY A$(J)
60 OUT &H1C00,0:A=INP(&H1B00):OUT &H1C00,1:B=INP(&H1B00)
70 POKE ADR,A,B:ADR=ADR+2:NEXT:NEXT:I=0
80 DATA C1,#C1,D1,#D1,E1,F1,#F1,G1,#G1,A1,#A1,B1
90 '-------------------------------- MML SET
100 ADR=0
110 READ A$:IF A$="*" THEN 140 ELSE IF A$="END" THEN 150
120 FOR J=1 TO LEN(A$):OUT &H3FFF+I*256+ADR+J,ASC(MID$(A$,J,1))
130 NEXT:ADR=ADR+LEN(A$):GOTO110
```

```
140 I=I+1:GOTO100
150 POKE &HFB49, 4,7: ' Turbo:160,31   RS-232C:168,31   ﾘｯﾀｲ ﾎﾞｰﾄﾞ:4,10
160 RUN "SP SUB.Bas"
170 '------------------ MUSIC 1
180 DATA "L1O3V9M7[2E6EEE5A5E9D6DDD5G5D9C6CCC5F5C9<B6BBB5B5B9>]
190 DATA "<B7>B7L5AGfED7C<B>C9C7DCE7GAB9R9
200 DATA "B7>C<BAGfED7C<B>C9CCDCE7GAB9R9:",*
210 '
220 DATA "L3O2V8M3([@A][@G][@F][@E]|)([@E][@D][@C]<[@B]>|):",*
230 '----------------- MUSIC 2
240 DATA "L3O2V9M5([4CdG]CdDG|)([4Fg>C<]FgG>C<|):",*
250 '
260 DATA "L3O1M3([4CCV7PCPV9]CCCC|)([4FFV7PFPV9]FFFF|):",*
270 '----------------- TENUKI
280 DATA *,*,*,*
290 '----------------- MUSIC 3
300 DATA "M9V7O3L3 [3(A>AGA<|) (G>GFG<|) (F>FEF<|) (G>GFG<|)]
310 DATA "(A>AGA<|) (G>GFG<|) (F>FEF<|) G>GFG<G>GFE
320 DATA "L5D8C <B>CDC <B8A G8A3B3 A7G6F3 G7A5B5 >C8E5 G9
330 DATA "ARL3RAGF G8FE A5R5RAGF G7E5DC <B7>C5D5 C8<BA G8AB A7AB>CD
340 DATA "L5C7DE F7ED <B7>CD C9 L9[〒RRR]:",*
350 '
360 DATA "M9V9O2L9RRRR AGFG L7AAGGFFGG M6L5AAAAGGGGFFFFGGL3GGGG
370 DATA "L5M5CCCC<BBBBAAAAGGGGFFFFGGGGAAAAL3[8B]
380 DATA ">(CCR5CCCC<GGGGAA|BB>)GG>L5CCCC<BBBBAAAA>L3[8C]
390 DATA "<[7B]A[7G]A[8B]C9 L9[〒RRR]:",*
400 DATA END
```

リスト3 MUSIC DRIVER.Bin

```
F800 F3 DD E5 F5 C5 D5 E5 DD : 06
F808 21 6A FA 21 4E FB AF 32 : D0
F810 67 FA CD 4C F8 DD 21 79 : E9
F818 FA 21 4E FC 3E 02 32 67 : 3E
F820 FA CD 4C F8 3A 48 FB FE : 86
F828 FF 20 18 3A 4D FB C6 03 : 82
F830 32 4D FB 30 0E 21 76 FA : 49
F838 35 20 01 34 21 85 FA 35 : 5F
F840 20 01 34 E1 D1 C1 F1 DD : 96
F848 E1 FB ED 4D DD 7E 0B F6 : 72
F850 00 CC 7B F8 DD 35 0B 3A : 96
F858 67 FA CB 3F C6 08 4F DD : 65
F860 7E 0D 06 1C ED 49 05 ED : D5
F868 79 DD 34 0E DD 7E 0E DD : DE
F870 BE 0A C0 DD 36 0E 00 DD : 86
F878 35 0D C9 DD 5E 00 DD 56 : 79
-----------------------------------
SUM: 27 7F 84 3D AE E9 5E 06 94E0

F880 01 E5 19 CD B7 F9 FE 0D : 87
F888 30 0B F5 CD CC F9 32 69 : 5D
F890 FA F1 C3 61 F9 20 09 CD : FE
F898 CC F9 DD 77 03 E1 18 E1 : F6
F8A0 FE 0F 30 0A 7E D6 30 DD : A8
F8A8 77 02 13 E1 18 D3 20 0F : 87
F8B0 7E D6 30 87 DD 77 0A 13 : 7C
F8B8 DD 36 0E 00 C3 5D F9 FE : 38
F8C0 13 30 0C D6 11 DD 46 02 : 5B
F8C8 80 DD 77 02 C3 5D F9 20 : 0F
F8D0 0D DD 73 04 DD 72 05 DD : 92
F8D8 36 06 00 C3 5D F9 FE 15 : 68
F8E0 30 0F DD 7E 06 F6 00 28 : BE
F8E8 74 DD 5E 04 DD 56 05 18 : 03
F8F0 6C 20 14 62 6B DD 5E 04 : AC
F8F8 DD 56 05 DD 75 04 DD 74 : DF
-----------------------------------
SUM: 8A 49 79 44 86 42 26 ED D039

F900 05 DD 36 06 01 18 56 FE : 8B
F908 17 30 0F 13 DD 73 07 DD : 9D
F910 72 08 7E D6 30 DD 77 09 : 5B
F918 18 43 20 11 DD 7E 09 3D : 2D
F920 28 3B DD 77 09 DD 5E 07 : 02
F928 DD 56 08 18 30 FE 19 30 : CA
F930 13 3A 48 FB FE FF 28 09 : BE
F938 7E D6 2A DD 77 0C DD 77 : 32
F940 0D 13 18 19 20 12 3A 4C : 09
F948 FB EE 12 32 4C FB 01 07 : 7C
F950 1C ED 49 05 ED 79 18 05 : DA
F958 11 00 00 18 00 E1 C3 81 : 4E
F960 F8 F6 00 20 0A D5 11 00 : FE
F968 00 CD A5 F9 D1 18 1E 26 : 98
F970 00 DD 6E 02 29 29 29 44 : 0C
F978 4D 29 09 01 88 FA 09 3D : 48
-----------------------------------
SUM: B6 B0 C9 EB 7E 43 D0 58 6A66

F980 87 4F 06 00 09 D5 56 23 : 33
F988 5E CD A5 F9 D1 DD 7E 0C : 01
F990 DD 77 0D DD 36 0E 00 3A : BC
F998 69 FA DD 77 0B DD 73 00 : 12
F9A0 DD 72 01 E1 C9 06 1C 3A : 56
F9A8 67 FA ED 79 05 ED 51 3C : 46
F9B0 04 ED 79 05 ED 59 C9 7E : FC
F9B8 23 06 1B 4F E5 21 5C FA : EF
F9C0 7E B9 28 03 2B 10 F9 78 : 0E
F9C8 3D E1 13 C9 7E 23 D6 30 : A1
F9D0 FE 0A 38 04 DD 7E 03 C9 : 6B
F9D8 26 00 6F 01 5D FA 09 7E : 74
F9E0 13 C9 F3 ED 4B 49 FB 3E : 89
F9E8 03 16 04 C5 ED 79 03 15 : 60
F9F0 20 FA C1 21 00 F8 22 5E : 74
F9F8 00 3E 58 ED 79 03 03 03 : 05
-----------------------------------
SUM: AB A7 09 8C 4F 72 D7 FA 0768

FA00 3E A7 ED 79 3A 4B FB ED : B8
FA08 79 3A 48 FB 87 01 00 40 : BE
FA10 80 47 26 02 11 4E FB 2E : 77
FA18 00 ED 78 03 12 13 2D 20 : DA
FA20 F8 25 20 F3 AF 32 67 FA : 72
FA28 21 67 FA 11 68 FA 01 20 : 16
FA30 00 ED B0 3E 1C 32 4C FB : 70
FA38 01 07 1C ED 49 05 ED 79 : C5
FA40 FB C9 52 43 63 44 64 45 : A9
FA48 46 66 47 67 41 61 42 4C : 8A
FA50 4F 4D 3C 20 3E 28 7C 29 : 03
FA58 5B 5D 56 50 3A 04 08 0C : B0
FA60 10 18 20 30 40 60 80    : 98
-----------------------------------
SUM: 4C 86 04 F2 BC 41 6E CF 1626
```

リスト4 SP SUB.Bas

```
10 'POKE &HF9E2,&HC9
20 IF MEM$(&HD000,3)=HEXCHR$("ED 73 D4") THEN 40
30 CLEAR &HCFFF:LOADM "SP MAIN.Bin":LOADM"ROUND DATA.Bin"
40 WIDTH40:SCREEN:POKE &HFB4B,140
50 DEFINTA-Z:FOR I=6 TO 12:READ A: SOUND I,A:NEXT:DATA 30,28,0,0,16,0,35
60 DIM A$(11),B$(1),TITLE$(7)
70 '------------------------------------------ SET TITLE
80 LP=7:L(0)=4:L(1)=4:B$(0)=" ":B$(1)="ｷ":"MAKE"
90 FOR I=0 TO 7:TITLE$(I)=A$(I):NEXT
100 LP=11:L(0)=2:L(1)=3:B$(0)=CHR$(28):B$(1)="ﾛ":"MAKE"
110 FOR I=0 TO 3:READ A$:F$(I)=HEXCHR$(STRING$(4,A$)+STRING$(32,"0")):NEXT
120 '------------------------------------------ TITLE
130 CGEN1:FOR I=1 TO 32:FOR J=0 TO 6 STEP 2
140 LOCATE5,J+5:PRINT RIGHT$(TITLE$(J),I);
150 LOCATE37-I,J+6:PRINT LEFT$(TITLE$(J+1),I);
160 NEXT:NEXT
170 DEFCHR$(219)=STRING$(24,CHR$(0))
180 FOR I=0 TO 11:LOCATE9,3+I:PRINTA$(I);:NEXT
190 A$=STRING$(8,CHR$(0)):B$=STRING$(16,CHR$(0))
200 FOR I=0 TO 3:DEFCHR$(219)=F$(I):PAUSE1:NEXT
210 '
220 RD=0:SP=4:CGEN
230 LOCATE8,20:PRINT"Press Trigger To Start!"
240 LOCATE8,18:PRINTUSING"ROUND:#          SPEED:#",RD+1,SP+1
250 S=STICK(1):RD=RD+(S=4ANDRD>0)-(S=6ANDRD<3)
260 SP=SP+(S=2ANDSP>0)-(S=8ANDSP<8)
270 PAUSE1:IF STRIG(1)=0 THEN 240
280 MEM$(&HDCEA,9)="ｱｱｱﾊﾊﾊﾊﾊﾊ"
290 POKE &HDF37,1,SP*5,85,48,85 :POKE &HE084,0,4,RD
300 CGEN1:FORI=0TO1:SCREEN,I:LINE(0,0)-(39,24)," ",BF:NEXT
310 FOR I=0 TO 1000 STEP 128:MEM$(&HE087+I,128)=STRING$(128,CHR$(0)):NEXT
320 SCREEN:CALL &HD000:POKE &HFB48,255
330 IF PEEK(&HE085)<>255 THEN 380
340 SCREEN0,0:SCREEN:LOCATE12,11:PRINT"<< GAME OVER >>"
350 PAUSE70:CLS:GOTO130
360 END
370 '------------------------------------- ENDING
380 SCREEN0,1:CLS4:YA$=CHR$(31)+CHR$(29)+CHR$(29)+CHR$(29)
390 FOR I=0 TO 100:PSET(RND(1)*287+16,RND(1)*199,RND(1)*2+1):NEXT
400 LINE(0,0)-(15,199),PSET,4,BF:LINE(303,0)-(319,199),PSET,4,BF
410 FOR I=0 TO 360 STEP 10
420 LINE(20,9)-(20+COS(RAD(I))*7,9+SIN(RAD(I))*6),"ｯ",BF:NEXT
430 SCREEN 1,1:PALET 4,0:PALET 2,5:PALET 3,7:PRW 16
```

```
440 FOR I=0 TO 100:LINE(161-I,99+I)-(160-I,100+I),XOR,7
450 FOR J=0 TO 100:NEXT:NEXT :PAUSE20
460 FOR I=0 TO 37:LOCATEI,19:PRINT"abc"+YA$+"def";:FOR J=0 TO 100:NEXT
470 LOCATEI,19:PRINT"   "+YA$+"   ";:NEXT
480 C(0)=0:C(1)=2:C(2)=6:C(3)=7 :SOUND 12,50
490 FOR I=0 TO 3:PALET0,C(I):PALET4,C(I):SOUND13,0:FOR J=0 TO 10:NEXT:NEXT
500 CLS:FOR I=0 TO 15:SOUND13,0:PAUSE1:NEXT
510 PAUSE 20:FOR I=2 TO 0 STEP -1:PALET0,C(I):PALET4,C(I):PAUSE 3:NEXT
520 PAUSE20:POKE &HFB48,4:POKE &HFB4B,255:CALL &HF9E2:PAUSE50
530 RESTORE 760 :P=10 :CGEN
540 READ A$:IF A$="END" THEN 570
550 LOCATE 20-LEN(A$)/2,24:PRINT A$:PAUSE P :FOR I=0 TO 1:PRINT:PAUSE P :NEXT
560 GOTO540
570 FOR I=0 TO 6:PRINT:PAUSE P :NEXT
580 LOCATE0,9 :PRINT CHR$(14):PAUSE P :GOTO580
590 END
600 '--------------------------- DATA ﾃﾝｶｲ (L(n)=1 ｷﾞｮｳﾉ LENGTH,LP=LOOP ｶｲｽｳ)
610 LABEL"MAKE":READ A$,B$:B$=A$+B$:L=L(0)+L(1)
620 FOR I=0 TO LP:A$="":D$=MID$(B$,I*L+1,L)
630 FOR J=0 TO 1:A=VAL("&H"+MID$(D$,L(0)*J+1,L(J)))
640 A$=A$+RIGHT$(STRING$(L(J)*4,"0")+BIN$(A),L(J)*4):NEXT
650 FOR J=1 TO LEN(A$):MID$(A$,J,1)=B$(VAL(MID$(A$,J,1))):NEXT
660 A$(I)=A$:NEXT:RETURN
670 '--------------------------- TITLE DATA (Defeat)
680 DATA "7E00C0002101000020810000020878008"
690 DATA "4132319C417A7A4842424248FC3231A8"
700 '--------------------------- TITLE DATA (X)
710 DATA "FFBFF7FD021FE040FE4807EB007840"
720 DATA "00DC00858012DB8205CC40BFEFFDFF"
730 '--------------------------- FADE IN DATA
740 DATA "5500","55AA","FFAA","FFFF"
750 '--------------------------- STAFF ROLE
760 DATA "--- Defeat ---","the resist IV","Main Program","H.Asano"
770 DATA "Music composed","H.Kamon","J.Nakamoto","H.Ohgata"
780 DATA "Chr. design","H.Asano","Special thanks to","K.Kozima","S.Ishii"
790 DATA "T.Araki","","","Defeat -resist IV- in 1989"
800 DATA "(C)Kitasenri science club.",END
```

リスト5 SP MAIN.Bin

```
D000 ED 73 0D E4 3E 50 21 4A : 4A
D008 E1 77 11 4B E1 01 DF 00 : 75
D010 ED B0 21 80 ED 36 20 21 : A2
D018 FD DD 22 93 E0 21 0E 12 : B0
D020 22 87 E0 21 01 00 22 91 : 5E
D028 E0 AF 32 8D E0 06 2A C5 : 23
D030 CD 90 DA C1 10 F9 21 0E : 30
D038 12 01 E4 DC 11 03 02 CD : B6
D040 61 DA CD 7E DC 21 00 00 : 83
D048 CD 4C D2 CD 4C D2 06 B4 : 90
D050 C5 CD 90 DA 21 0E 12 01 : 3E
D058 E4 DC 11 03 02 CD 61 DA : DE
D060 CD 7E DC C1 10 EA AF 32 : C3
D068 48 FB CD E2 F9 3A 86 E0 : 8B
D070 21 00 E5 84 67 4E 06 00 : 45
D078 11 38 DE 23 ED B0 11 68 : 60
---------------------------------
SUM: B7 BE DD FF 96 9A 62 B7 58A6

D080 E0 0E 1A ED B0 11 3C DF : D1
D088 7E 23 4F E6 0F 28 1A 47 : 6E
D090 79 E6 F0 1F 1F 1F 1F E5 : B0
D098 21 D9 DC D5 5F 16 00 19 : 39
D0A0 D1 7E 12 13 10 FC E1 18 : 79
D0A8 DF 62 6B 3A 69 E0 4F ED : 6B
D0B0 42 ED 42 C5 ED B0 C1 ED : 81
D0B8 42 ED 42 C5 01 3C DF E5 : 37
D0C0 ED 42 E1 C1 30 ED 21 01 : 10
D0C8 FF 22 82 E0 21 38 DE 22 : DC
D0D0 93 E0 21 8E E0 36 00 23 : 5B
D0D8 36 00 3E 55 32 3B DF 06 : 1B
D0E0 01 CD 2E D2 3A 80 E0 F6 : 5E
D0E8 00 28 F4 3E FF 32 48 FB : CE
D0F0 06 64 CD 2E D2 3E 01 32 : A8
D0F8 48 FB CD E2 F9 06 AF 21 : C1
---------------------------------
SUM: 30 42 B4 42 0B C2 FB 8B 439A

D100 18 20 22 80 E0 3E 01 32 : 2B
D108 0A E4 C5 CD 39 D2 21 0A : B6
D110 E4 CB 0E 2A 80 E0 30 04 : 7B
D118 25 22 80 E0 ED 5B 69 E0 : 38
D120 01 3C DF CD 61 DA CD 7E : 6F
D128 DC C1 10 DE 3E FF 32 3B : 35
D130 DF 3E 55 32 0A E4 CD 39 : 98
D138 D2 CD E9 D7 CD 7E DC 3A : C0
D140 6E E0 FE FA 38 F0 3E FF : AB
D148 32 48 FB 3A 7C E0 CD F4 : CC
D150 D9 3A 7D E0 CD F4 D9 3E : 48
D158 C8 32 0A E4 CD 39 D2 ED : AD
D160 5B 69 E0 2A 80 E0 06 02 : 36
D168 E5 D5 ED 5F 15 92 BA 30 : 97
D170 FC 84 67 ED 5F 1D 93 BB : 9E
D178 30 FC 85 6F CD D1 D9 D1 : 68
---------------------------------
SUM: 66 4B DB E8 0B E3 45 28 E88A

D180 E1 10 E5 01 3C DF CD 61 : 20
D188 DA CD 7E DC 21 0A E4 34 : 44
D190 3E 0F BE 30 05 CD C3 D9 : A9
D198 18 C2 20 C0 36 00 2A 80 : 9A
D1A0 E0 24 22 80 E0 7C FE 1E : 1E
D1A8 38 B2 06 64 CD 2E D2 21 : 42
D1B0 86 E0 34 7E FE 04 DA 66 : 5A
D1B8 D0 21 1A DE 22 93 E0 21 : 9F
D1C0 8E E0 36 00 23 36 00 21 : 1E
D1C8 8A E0 36 0A 21 90 01 E5 : 41
D1D0 CD 90 DA 2A 87 E0 7C FE : 42
D1D8 0F 28 05 38 02 25 25 24 : E4
D1E0 7D FE 0F 28 05 38 02 2D : 1E
D1E8 2D 2C 22 87 E0 11 03 02 : F8
D1F0 01 E4 DC CD 61 DA CD 7E : 14
D1F8 DC CD 49 D2 E1 2B 7C B5 : 01
---------------------------------
SUM: FA D8 58 C7 59 10 18 3E 10FC

D200 20 CD 06 64 C5 CD 90 DA : 53
D208 2A 87 E0 11 03 02 01 E4 : 8C
D210 DC E5 CD 61 DA E1 2C 3E : 14
D218 2C BD 30 01 2D 22 87 E0 : D0
D220 CD 7E DC CD 49 D2 C1 10 : E0
D228 DB ED 7B 0D E4 C9 C5 CD : 8F
D230 39 D2 CD 7E DC C1 10 F6 : F9
D238 C9 CD 90 DA CD 52 D2 CD : BE
D240 50 D6 CD F7 D4 CD 49 D2 : A6
D248 C9 2A 37 DF 2B 7C B5 20 : 85
D250 FB C9 21 8C E0 7E FE 50 : 1D
D258 38 14 35 3D FE 51 C2 AF : 7E
D260 D3 3A 85 E0 FE FF C2 AF : E0
D268 D3 ED 7B 0D E4 C9 F3 3E : 26
D270 0E 01 00 1C ED 79 05 ED : 83
D278 78 FB 4F 2A 87 E0 E5 3A : 72
---------------------------------
SUM: 74 00 40 DB D8 B9 09 81 0652

D280 8C E0 F6 00 20 43 25 CD : B7
D288 FA D3 28 02 CB C1 24 24 : CB
D290 24 CD FA D3 28 02 CB C9 : 7C
D298 25 25 2D CD 0E D4 28 02 : 50
D2A0 CB D1 2C 2C CD 0E D4 28 : CB
D2A8 10 C5 CD C3 D9 E5 21 85 : C9
D2B0 E0 35 E1 3E 70 32 8C E0 : 42
D2B8 C1 2C CD 0E D4 28 02 CB : 91
D2C0 91 2C CD 0E D4 28 02 CB : 61
D2C8 D9 79 E1 11 1C 05 01 2A : 90
D2D0 04 CD D2 D4 22 87 E0 1F : 1F
D2D8 1F F5 30 06 AF 32 89 E0 : 94
D2E0 18 22 3A 89 E0 F6 00 20 : F3
D2E8 17 3A 8B E0 87 4F 06 00 : 98
D2F0 E5 21 74 DD 09 7E 23 66 : 67
D2F8 6F 7A CD 95 D4 E9 3E 08 : 4E
---------------------------------
SUM: 5B FA A2 B1 10 B9 92 96 A2D3

D300 3D 32 89 E0 F1 1F 30 03 : 1B
D308 AF 18 16 3A 8A E0 F6 00 : 77
D310 20 0F 3A 8B E0 3C 32 8B : CD
D318 E0 D6 05 38 04 32 8B E0 : 94
D320 3C 32 8A E0 3A 8C E0 01 : 7F
D328 E4 DC F6 00 28 0B 3D 32 : 58
D330 8C E0 CB 57 28 03 01 F3 : AD
D338 DC 11 03 02 CD 61 DA 3A : 34
D340 8C E0 F6 00 28 06 3D 32 : FF
D348 8C E0 18 63 2A 87 E0 DD : 55
D350 21 62 E1 06 14 11 08 00 : 97
D358 DD 7E 00 95 FE 03 30 08 : 29
D360 DD 7E 01 94 FE 02 38 3B : 63
D368 DD 19 10 EC DD 21 2A E2 : FC
D370 06 14 DD 7E 04 F6 00 28 : 97
D378 21 DD 5E 01 7B 95 C6 02 : 35
---------------------------------
SUM: 6B 56 67 13 74 B7 58 2C 4023

D380 FE 05 30 16 DD 56 02 7A : F8
D388 94 3C FE 03 30 0C E5 EB : DD
D390 CD C3 D9 E1 DD 36 04 00 : 61
D398 18 09 11 0B 00 DD 19 10 : 43
D3A0 D1 18 0C CD C3 D9 3E 70 : 0C
D3A8 32 8C E0 21 85 E0 35 DD : 36
D3B0 21 4A E1 06 06 DD 6E 00 : A3
D3B8 DD 66 01 3E 2B BD 38 2E : D0
D3C0 3E 1E BC 38 29 3E 03 BC : 76
D3C8 30 24 BD 30 21 DD 7E 02 : BF
D3D0 85 6F DD 7E 03 84 67 DD : 1A
D3D8 75 00 DD 74 01 CD C6 DC : 36
D3E0 E5 CD 75 DC 7E FE 21 E1 : 81
D3E8 30 04 36 21 18 04 DD 36 : BA
D3F0 00 64 11 04 00 DD 19 10 : 7F
D3F8 BC C9 E5 06 03 CD C6 DC : E2
---------------------------------
SUM: B1 10 BA 98 4A E0 A8 6A 5F29

D400 CD 75 DC 7E FE 20 20 04 : DE
D408 23 10 F8 AF E1 C9 E5 06 : 6F
D410 02 CD C6 DC CD 75 DC 11 : A0
D418 30 00 7E FE 20 20 04 19 : 09
D420 10 F8 AF E1 C9 E1 11 02 : 55
D428 00 2C CD AB D4 2D C3 FE : 66
D430 D2 E1 FE 02 DA FE D2 11 : 6E
D438 02 00 25 2C CD AB D4 24 : C3
D440 CD AB D4 2D C3 FE D2 E1 : ED
D448 FE 03 DA FE D2 11 02 00 : BE
D450 2C CD AB D4 16 FF 2C CD : 86
D458 AB D4 16 01 CD AB D4 2D : 0F
D460 2D C3 FE D2 E1 FE 02 DA : 7B
D468 FE D2 11 00 02 2C 25 CD : 01
D470 AB D4 16 FE 24 CD AB D4 : 03
D478 2D C3 FE D2 E1 CD 95 D4 : D7
---------------------------------
SUM: AB D2 49 63 70 B2 9A 93 2E48

D480 FE 02 DA FE D2 11 02 00 : BD
D488 2C CD AB D4 2D 1E FE CD : 8E
D490 AB D4 C3 FE D2 E5 21 4A : 62
D498 E1 01 00 06 11 04 00 7E : 7B
D4A0 FE 2C 38 01 0C 19 10 F7 : 8F
D4A8 E1 79 C9 D5 11 04 00 DD : EA
D4B0 21 4A E1 06 06 DD 7E 00 : B3
D4B8 FE 2B 30 06 DD 19 10 F5 : 5A
D4C0 D1 C9 DD 75 00 24 DD 74 : 61
D4C8 01 25 D1 DD 73 02 DD 72 : 98
D4D0 03 C9 1F 38 01 25 1F 38 : A0
D4D8 01 24 1F 38 01 2D 1F 38 : 01
D4E0 01 2C F5 7C BA 30 01 24 : AD
D4E8 BB 38 01 25 7D B8 30 01 : 7F
D4F0 2C B9 38 01 2D F1 C9 21 : 26
D4F8 3B DF CB 0E 30 55 FD 21 : 96
---------------------------------
SUM: AD 95 3F 2A EB D1 AE 1B D833

D500 62 E1 06 14 11 08 00 FD : 73
D508 7E 02 F6 00 28 09 FD 4E : F2
D510 06 81 FD 77 06 30 12 FD : 40
D518 7E 00 FD 4E 04 81 FD 77 : C2
D520 00 FE 04 38 26 FE 2C 30 : BA
D528 22 FD 7E 03 F6 00 28 09 : C7
D530 FD 4E 07 81 FD 77 07 30 : 7E
D538 16 FD 7E 01 FD 4E 05 81 : 63
D540 FD 77 01 FE 04 38 04 FE : B1
D548 1D 38 04 FD 36 00 32 FD : BB
D550 19 10 B4 FD 21 62 E1 06 : 44
D558 14 FD 7E 00 FE 2C 30 19 : 02
D560 6F FD 66 01 CD C6 DC E5 : 27
D568 CD 75 DC 7E E1 FE 21 38 : D4
D570 06 FD 36 00 32 18 02 36 : BB
D578 23 11 08 00 FD 19 10 D9 : 3B
---------------------------------
```

▶８月号のX1特集はすごくよかったです。僕自身も「ちょっとなんかする」というときにはX68000よりX1のほうが使い勝手がよくていいな，と思うことがしばしばあります。X1はほかのマシンと違って質のいいユーザーが多いはずですから(MZも)これからもずっと生き延びてゆくでしょう。要はユーザーの気合ですよね。　小田島　浩一（17）東京都

```
SUM: 45 E6 B4 0D 8F 40 C2 EF EC04

D580 C9 B7 ED 5B 87 E0 14 1C : 5F
D588 E5 ED 52 E1 C8 D5 CD E0 : 4F
D590 D5 D1 D8 FD 75 00 FD 74 : 61
D598 01 7B 95 CD D5 D5 FD 70 : F5
D5A0 04 F5 7A 94 CD D5 D5 FD : 7B
D5A8 70 05 4F F1 B9 38 10 61 : 17
D5B0 2E 00 4F CD F5 D5 FD 36 : 47
D5B8 02 00 FD 75 03 18 0D 67 : 03
D5C0 2E 00 CD F5 D5 FD 75 02 : 39
D5C8 FD 36 03 00 FD 36 06 00 : 6F
D5D0 FD 36 07 00 C9 06 00 C8 : D1
D5D8 38 02 04 C9 ED 44 05 C9 : 06
D5E0 FD 21 62 E1 11 08 00 06 : 80
D5E8 14 FD 7E 00 FE 2B D0 FD : 85
D5F0 19 10 F6 37 C9 AF 06 10 : E4
D5F8 29 17 2C 91 30 02 2D 81 : DD
--------------------------------
SUM: DB 9D 9E 34 A7 E5 4D 02 1893

D600 10 F6 C9 7C E6 0F 4F DD : 6C
D608 21 06 E3 06 14 DD 7E 00 : 7F
D610 F6 00 28 08 11 05 00 DD : 19
D618 19 10 F2 C9 E5 7C F5 DD : 17
D620 36 04 00 E6 80 28 04 DD : A9
D628 36 04 01 F1 E6 70 0F 0F : A0
D630 0F 5F 16 00 21 7E DD 19 : 19
D638 5E 23 56 EB DD 75 02 DD : F3
D640 74 03 E1 DD 75 01 79 07 : 2B
D648 07 DD 77 00 0D 20 B8 C9 : 09
D650 DD 21 06 E3 06 14 11 05 : 17
D658 00 C5 D5 DD 7E 00 F6 00 : EB
D660 28 50 DD 35 00 20 4B FD : F2
D668 21 2A E2 06 14 11 0B 00 : 63
D670 FD 7E 04 F6 00 28 06 FD : A0
D678 19 10 F5 18 35 FD 36 00 : 9E
--------------------------------
SUM: D0 64 1E FB A3 83 7E 48 FC79

D680 00 FD 36 01 2D DD 7E 01 : BD
D688 FD 77 02 DD 6E 02 DD 66 : 06
D690 03 4E 23 46 23 FD 71 08 : 53
D698 FD 70 09 FD 75 05 FD 74 : 5E
D6A0 06 FD 75 03 FD 74 04 ED : DD
D6A8 5F FD 77 07 DD 7E 04 FD : 36
D6B0 77 0A D1 C1 DD 19 10 A1 : BA
D6B8 DD 21 2A E2 06 14 C5 DD : C6
D6C0 6E 03 DD 66 04 7C F6 00 : 2A
D6C8 CA 4D D7 DD 7E 00 F6 00 : 3F
D6D0 20 20 7E FE FF 20 07 DD : BF
D6D8 6E 05 DD 66 06 7E CB BF : C4
D6E0 CB 3F CB 3F CB 3F CB 3F : 28
D6E8 DD 77 00 23 DD 75 03 DD : A9
D6F0 74 04 DD 35 00 2B 7E F5 : 28
D6F8 DD 6E 01 DD 66 02 DD 46 : B4
--------------------------------
SUM: 75 F4 03 E9 85 FB 8D 3E EA9B

D700 0A 05 20 02 EE 03 01 50 : 73
D708 00 11 22 01 CD D2 D4 F1 : 98
D710 CB 7F 28 03 CD D2 D4 3E : 26
D718 01 BD 38 06 DD 36 04 00 : 13
D720 18 2B DD 75 01 DD 74 02 : E9
D728 DD 34 07 3A 68 E0 DD BE : 35
D730 07 30 09 DD 36 07 00 E5 : 3F
D738 CD 81 D5 E1 CD B1 D7 38 : 91
D740 0C DD 4E 08 DD 46 09 11 : 7C
D748 03 02 CD 61 DA 11 0B 00 : 29
D750 DD 19 C1 05 C2 BE D6 DD : EF
D758 21 A6 E3 06 14 C5 DD 6E : D4
D760 00 DD 66 01 7D FE 03 38 : FA
D768 40 DD 4E 02 DD 46 03 11 : A4
D770 02 02 E5 CD 61 DA E1 CD : 9F
D778 B1 D7 30 07 3E 01 CD F4 : BF
--------------------------------
SUM: 9F 93 EC C4 57 4B 50 C2 7DDE

D780 D9 18 26 3A 92 E0 47 3A : 44
D788 91 E0 3D B8 20 08 DD 35 : A0
D790 00 20 03 DD 34 00 DD 34 : 45
D798 04 3A 68 E0 DD BE 04 30 : 55
D7A0 08 DD 36 04 00 24 CD 81 : 91
D7A8 D5 11 05 00 DD 19 C1 10 : B2
D7B0 AC 7D FE 2C D0 FD 21 4A : 8B
D7B8 E1 06 06 11 04 00 FD 7E : 7D
D7C0 00 95 FE 03 30 08 FD 7E : 49
D7C8 01 94 FE 02 38 05 FD 19 : E8
D7D0 10 EC C9 FD 36 00 64 DD : 39
D7D8 36 04 00 DD 36 00 01 CD : 1B
D7E0 C3 D9 3E 01 CD F4 D9 37 : AC
D7E8 C9 21 7E E0 7E 23 34 BE : DB
D7F0 2A 80 E0 ED 5B 69 E0 20 : 3B
D7F8 1C AF 32 7F E0 3A 73 E0 : E9
--------------------------------
SUM: F1 05 A0 1C CE A7 70 62 93E0

D800 FE 0D 38 11 3A 6B E0 1F : F8
D808 F5 DC BE D8 F1 1F F5 DC : 48
D810 D3 D8 F1 18 05 3A 6B E0 : 3E
D818 1F 1F E5 D5 1F F5 DC E8 : D0
D820 D8 F1 1F F5 DC 0D D9 F1 : 90
D828 1F F5 DC 52 D9 F1 D1 E1 : BE
D830 22 80 E0 E5 01 3C DF CD : 50
D838 61 DA E1 ED 5B 6C E0 19 : C9
D840 DD 21 4A E1 06 06 C5 DD : D7
D848 7E 00 95 FE 03 30 1B DD : 3C
D850 7E 01 94 FE 03 30 13 E5 : 3C
D858 DD 6E 00 DD 66 01 CD C3 : 1F
D860 D9 DD 36 00 64 21 6E E0 : BF
D868 35 E1 11 04 00 DD 19 C1 : E2
D870 10 D4 3A 8C E0 F6 00 20 : A0
D878 18 2A 87 E0 0E 02 06 03 : C2
--------------------------------
SUM: 4B 6C 03 19 24 BC D2 A1 594F

D880 E5 CD C6 DC 7E FE B1 30 : B1
D888 24 23 10 F8 E1 24 0D 20 : 81
D890 ED 21 4A E1 06 06 5E 23 : C6
D898 56 2B EB CD C6 DC EB 1A : E0
D8A0 FE B1 38 02 36 64 11 04 : 98
D8A8 00 19 10 EA C9 E1 2A 87 : 6E
D8B0 E0 CD C3 D9 21 85 E0 35 : 04
D8B8 3E 70 32 8C E0 C9 3A 82 : D1
D8C0 E0 47 85 6F FE 08 38 04 : 5D
D8C8 83 FE 2C D8 78 ED 44 32 : 60
D8D0 82 E0 C9 3A 83 E0 47 84 : 93
D8D8 67 FE 05 38 04 82 FE 1E : 44
D8E0 D8 78 ED 44 32 83 E0 C9 : DF
D8E8 21 6F E0 34 23 7E 2B BE : 2E
D8F0 D0 36 00 2A 80 E0 3E 05 : D3
D8F8 85 6F 3A 71 E0 84 67 E5 : 4F
--------------------------------
SUM: 02 F2 CE 9F DD 53 CD 18 39C1

D900 CD 81 D5 E1 3A 72 E0 84 : 14
D908 67 CD 81 D5 C9 21 73 E0 : C7
D910 35 7E FE 05 28 2D F6 00 : 01
D918 20 10 ED 5F 6F 3A 74 E0 : 79
D920 4F C5 CD F5 D5 C1 81 32 : 1F
D928 73 E0 3A 75 E0 FE 03 D8 : BB
D930 2A 75 E0 01 2A DD 2D 2D : E1
D938 2D 22 75 E0 11 03 07 CD : 8C
D940 61 DA C9 2A 6C E0 ED 5B : C2
D948 80 E0 19 25 25 22 75 E0 : 3A
D950 18 D8 21 77 E0 35 C0 23 : 80
D958 7E 2B 77 2A 79 E0 ED 5B : EB
D960 80 E0 19 EB 2A 87 E0 E5 : DA
D968 3A 7B E0 47 C5 E5 D5 CD : 28
D970 8A D9 84 67 CD 8A D9 85 : 03
D978 6F 22 87 E0 EB CD 81 D5 : 06
--------------------------------
SUM: CC 2B 1B CE 1B 73 93 0D 2F55

D980 D1 E1 C1 10 E7 E1 22 87 : F4
D988 E0 C9 ED 5F E6 1F D6 10 : E0
D990 C9 DD 21 6A E3 06 14 C5 : F3
D998 DD 7E 02 F6 00 28 1B 3D : D3
D9A0 07 07 6F 26 00 01 16 DD : 97
D9A8 09 44 4D DD 6E 00 DD 66 : 28
D9B0 01 11 02 02 CD 61 DA DD : FB
D9B8 35 02 C1 11 03 00 DD 19 : 02
D9C0 10 D5 C9 F3 C5 01 0D 1C : 90
D9C8 ED 49 01 00 1B ED 49 FB : 83
D9D0 C1 C5 FD 21 6A E3 06 14 : 0B
D9D8 11 03 00 FD 7E 02 F6 00 : 87
D9E0 28 06 FD 19 10 F5 C1 C9 : D3
D9E8 FD 75 00 FD 74 01 FD 36 : 17
D9F0 02 05 C1 C9 21 0C E4 86 : 28
D9F8 27 77 3E 00 2B 8E 27 77 : 33
--------------------------------
SUM: BA 40 13 D5 86 F3 EC F9 1D4E

DA00 3A 84 E0 BE D0 21 85 E0 : B2
DA08 34 C6 02 27 32 84 E0 C9 : 82
DA10 3A 3A DF 80 47 1A F6 00 : 2A
DA18 C8 ED 79 03 13 18 F6 01 : 53
DA20 00 00 11 3F DD CD 10 DA : E4
DA28 3A 0B E4 CD 4C DA 3A 0C : 62
DA30 E4 CD 4C DA AF CD 4C DA : 79
DA38 13 01 21 00 CD 10 DA 3A : 26
DA40 85 E0 FE FF 20 02 3E 00 : C2
DA48 CD 4C DA C9 F5 E6 F0 1F : A6
DA50 1F 1F 1F C6 30 ED 79 03 : BC
DA58 F1 E6 0F C6 30 ED 79 03 : 45
DA60 C9 CD C6 DC D5 E5 E5 D5 : AC
DA68 11 FF F5 ED 52 D1 E1 30 : 26
DA70 15 E5 CD 75 DC 7E E1 FE : 75
DA78 21 30 06 0A FE 21 38 01 : B9
--------------------------------
SUM: 13 5C 30 EA 77 72 C0 CD 87E5

DA80 77 03 23 1D 20 EB E1 11 : B7
DA88 30 00 19 D1 15 20 D5 C9 : ED
DA90 3A 91 E0 47 21 92 E0 34 : B9
DA98 7E 90 C2 66 DB 77 21 01 : AA
DAA0 E9 11 00 E9 01 80 04 ED : 55
DAA8 B0 21 8D E0 34 7E FE 03 : F1
DAB0 DA 66 DB 36 00 2A 93 E0 : EE
DAB8 3A 8F E0 F6 00 28 0A 3D : 0E
DAC0 32 8F E0 3A 90 E0 2B 18 : 8E
DAC8 06 7E FE FF CA 74 DB E5 : 7F
DAD0 11 2B E9 06 08 C5 21 EA : 03
DAD8 DC 17 38 03 21 F3 DC F5 : 13
DAE0 3E 03 01 03 00 ED B0 EB : CD
DAE8 01 2D 00 09 EB 3D 20 F2 : 71
DAF0 F1 C1 10 E1 E1 3A 8E E0 : 2C
DAF8 FE 01 28 66 23 7E E5 17 : 2A
--------------------------------
SUM: 5F 8C 5E 25 D8 52 9C CC 34D4

DB00 30 1E F5 11 2B 05 62 6B : 51
DB08 CD C6 DC CD 75 DC 7E FE : 09
DB10 20 28 05 14 14 14 18 EE : 8F
DB18 21 0E DD EB CD 4B DC F1 : DC
DB20 17 30 1E F5 11 2B 1B 62 : 13
DB28 6B CD C6 DC CD 75 DC 7E : 76
DB30 FE 20 28 05 15 15 15 18 : A2
DB38 EE 21 12 DD EB CD 4B DC : DD
DB40 F1 E1 F5 E6 7C 1F 1F F6 : 5D
DB48 00 28 09 5F 23 56 E5 EB : D9
DB50 CD 03 D6 E1 F1 17 30 0A : C9
DB58 23 5E 23 56 E5 EB CD 03 : 9A
DB60 D6 E1 23 22 93 E0 11 F0 : 70
DB68 F0 21 00 E9 01 B0 04 ED : 9C
DB70 B0 C3 BA DB 23 7E FE 02 : A9
DB78 30 05 32 80 E0 18 07 20 : 06
--------------------------------
SUM: 33 8C D7 72 6B 5F 46 09 2245

DB80 09 3E 01 32 8E E0 23 C3 : CE
DB88 C9 DA FE 04 30 07 23 7E : 7D
DB90 32 91 E0 18 F1 20 0D 23 : FC
DB98 7E 32 8F E0 23 7E 32 90 : 82
DBA0 E0 C3 CF DA FE 06 30 0C : 8C
DBA8 23 11 EA DC 01 09 00 ED : F1
DBB0 B0 C3 C9 DA AF 32 8E E0 : 65
DBB8 18 CC 06 03 11 96 E0 3E : B2
DBC0 05 90 32 95 E0 C5 D5 D5 : AB
DBC8 DD E1 06 0F C5 DD 7E 00 : F3
DBD0 57 FE 00 28 33 3A 95 E0 : 5F
DBD8 DD 46 02 80 FE 08 38 03 : E6
DBE0 D6 08 15 DD 77 02 6A DD : 90
DBE8 66 01 DD 75 00 2C 2C 2C : 3D
DBF0 2C 47 3A 95 E0 D6 02 CB : C5
DBF8 27 CB 27 CB 27 80 57 CD : AF
--------------------------------
SUM: F2 0E 83 BF E5 C4 32 64 F774

DC00 C6 DC 7E FE 21 30 01 72 : E2
DC08 11 03 00 DD 19 C1 10 BC : 97
DC10 E1 E5 06 0F 7E F6 00 28 : 77
DC18 05 19 10 F8 18 20 ED 5F : AA
DC20 FE 28 30 1A E5 21 39 DF : 8E
DC28 CB 06 E1 30 11 36 28 23 : 74
DC30 ED 5F E6 1F FE 18 30 06 : 9D
DC38 C6 05 77 23 36 00 D1 21 : 8D
DC40 3C 00 19 54 5D C1 05 C2 : 8E
DC48 BF DB C9 DD 21 A6 E3 06 : F0
DC50 14 D5 11 05 00 DD 7E 00 : 5A
DC58 FE 03 38 06 DD 19 10 F5 : 3A
DC60 D1 C9 D1 DD 75 00 DD 74 : 0E
DC68 01 DD 73 02 DD 72 03 ED : 92
DC70 5F DD 77 04 C9 D5 11 F0 : 56
DC78 07 B7 ED 52 D1 C9 CD 91 : F5
--------------------------------
SUM: 7E 5C D5 DF 41 E3 94 7D D098

DC80 D9 21 F4 F0 0E 28 3A 3A : 88
DC88 DF 47 F5 16 18 1E 28 7E : 0D
DC90 ED 79 23 03 1D 20 F8 3E : FF
DC98 08 85 6F 30 01 24 15 20 : 86
DCA0 EC CD 1F DA 01 19 00 3A : 06
DCA8 8B E0 87 87 87 11 4C DD : 3A
DCB0 83 5F CD 10 DA 01 00 18 : B2
DCB8 3E 0C ED 79 0C F1 ED 79 : 13
DCC0 EE 04 32 3A DF C9 C5 7D : 48
DCC8 6C 26 00 29 29 29 29 44 : 7A
DCD0 4D 29 09 06 F0 4F 09 C1 : 8E
DCD8 C9 20 B1 B2 BE B7 C2 B6 : 39
DCE0 B8 B9 BD B5 61 62 63 64 : 6D
DCE8 65 66 BE BE B1 B1 BE BE : 25
DCF0 BE BE BE 20 20 20 20 20 : DA
DCF8 20 20 20 20 6D 6E 6F 70 : 3A
--------------------------------
SUM: 50 EE 20 F1 07 3F 11 A8 7475

DD00 71 72 73 74 75 76 77 78 : A4
DD08 67 68 69 6A 6B 6C 90 92 : 9B
DD10 8C 8E 8B 8D 8F 91 87 89 : 62
DD18 88 8A 83 85 84 86 7F 81 : 24
DD20 80 82 7D 7B 7E 7C 79 7B : E8
DD28 7A 7C 20 20 C8 20 C8 B6 : 9C
DD30 C8 B6 20 C8 B6 20 C8 B6 : BA
DD38 20 20 C8 B6 20 20 C8 53 : 19
DD40 43 4F 52 45 3A 00 4C 45 : F4
DD48 46 54 3A 00 4E 4F 52 4D : 10
DD50 41 4C 20 00 54 57 49 4E : EF
DD58 20 20 20 00 55 50 2D 44 : 76
DD60 4F 57 4E 00 33 2D 57 41 : EC
DD68 59 20 20 00 42 45 48 49 : B1
DD70 4E 44 20 00 25 D4 31 D4 : B0
DD78 64 D4 47 D4 7C D4 8E DD : 0E
--------------------------------
SUM: 12 64 10 22 56 E5 50 AD F059

DD80 9D DD BA DD C9 DD D2 DD : 66
DD88 D6 DD DD DD ED DD FC DC : 0F
DD90 2B 1A 1B 4A 1B 1A 2B 19 : 23
DD98 1B 49 1B 19 FF 02 DD 7B : F1
DDA0 7B 3B 1A 1B 2A 1E 1A 2E : 7B
DDA8 16 1E 26 17 16 27 15 17 : DA
DDB0 25 1D 15 2D 19 1D 29 1B : FE
DDB8 19 FF 02 DD 7B 5B 3B 1A : 22
DDC0 1B 2A 1E 1A 3E 3B FB FB : EC
DDC8 FF FC DC FB CB 76 FB CB : D9
DDD0 75 FF 08 DD 7B FF 08 DD : B8
DDD8 7B 7B 7B 7A FF 02 DD 7A : 43
DDE0 1B 1A 2B 1A 7B DB 4B 19 : 34
DDE8 1B 29 1B 79 79 FC DC 6B : 94
DDF0 EE 3B E9 3A 6E DB FD AD : 3F
DDF8 76 36 FB FB FF FF 02 FF : A1
--------------------------------
SUM: 2C E6 CB 8D 88 F6 6A 14 DD10

DE00 03 01 FF 04 01 FF FF 04 : 0A
DE08 05 E7 FF 04 14 F7 E6 C6 : A6
DE10 86 06 FF 04 0C 04 FF 04 : A2
DE18 FF 00 FF 02 FF 03 04 FF : 05
DE20 04 06 C3 FF 03 03 FF 04 : D5
DE28 08 E7 FF 03 02 FF 04 0C : 02
DE30 EF FF 03 01 FF 04 FF 00 : F4
--------------------------------
SUM: 88 DA C1 11 24 03 EA DD A1DB
```

▶どなたかX1用のディスク版（あったかな）「黄金の墓」を譲ってくらはい。どこにもないのや。見つからへんのや。
岡村　直也（18）兵庫県

リスト6 ROUND DATA.Bin

```
E500 C9 FF 03 07 FF 04 07 00 : DC
E508 0F 04 00 00 3B 14 05 94 : FB
E510 00 00 00 36 45 0A 04 FF : 88
E518 05 C7 C7 C7 C7 C7 C7 C7 : 76
E520 C7 C7 FF 04 06 01 00 00 : 98
E528 6C 05 10 85 00 00 40 00 : 46
E530 03 00 07 40 0F 00 07 00 : 60
E538 03 40 01 0F 14 01 00 01 : 69
E540 00 01 40 03 00 03 2C 44 : B7
E548 0F 44 07 05 94 0F 40 3F : 81
E550 00 1F 00 07 40 0F 00 07 : 7C
E558 00 03 00 01 40 01 40 09 : 8E
E560 00 09 00 1D 00 1D 25 44 : AC
E568 08 44 0D 40 09 0B 06 01 : B4
E570 00 01 00 03 00 07 00 03 : 0E
E578 00 01 40 43 00 E7 C0 43 : 6E
-----------------------------------
SUM: 2D 8C 75 8F 8C 23 B5 79 4C21

E580 00 01 40 03 00 07 00 03 : 4E
E588 00 01 40 01 40 03 00 07 : 8C
E590 00 0F 0F 45 07 40 00 00 : AA
E598 07 45 95 0F 00 07 00 03 : FA
E5A0 00 01 00 01 00 01 2B 05 : 33
E5A8 0F 05 03 00 07 00 37 00 : 55
E5B0 F3 C0 3B 00 1B 00 33 40 : 7C
E5B8 21 40 21 40 01 00 00 00 : C3
E5C0 00 00 FF 02 FF 01 FF 04 : 04
E5C8 FF 00 62 10 0B 05 05 04 : 8A
E5D0 1E 00 0F 01 0A 32 00 00 : 6A
E5D8 00 00 00 00 00 00 50 50 : A0
E5E0 04 00 00 00 08 15 03 1F : 43
E5E8 11 09 33 09 23 01 43 32 : EF
E5F0 21 04 26 41 31 51 32 22 : 62
E5F8 07 61 42 53 03 00 00 00 : 00
-----------------------------------
SUM: 84 CA 8E 49 DD F1 61 1D 0476

E600 C2 FF 03 07 FF 04 0A 00 : D8
E608 13 05 0A 05 00 0F 05 00 : 3B
E610 0A 45 16 56 00 05 94 00 : 54
E618 27 B4 19 34 FF 05 B9 B9 : 9E
E620 B9 B9 B9 B9 B9 B9 B9 03 : 12
E628 00 37 00 F7 00 E3 C0 C1 : 92
E630 00 C1 00 C1 12 44 81 C0 : 19
E638 81 80 81 40 83 00 87 0B : D7
E640 D6 83 40 81 40 E1 2F 44 : AE
E648 11 44 C1 80 C1 40 99 00 : 30
E650 99 40 83 95 54 8F 40 9F : B3
E658 00 87 C0 C3 40 83 40 A1 : AE
E660 80 E5 00 E5 40 CD 80 C7 : 9E
E668 00 87 80 83 40 93 00 91 : EE
E670 40 9B 00 9B 00 8B 80 81 : 02
E678 40 81 40 81 80 81 80 C3 : C6
-----------------------------------
SUM: C0 44 7A 24 E1 9C A5 68 A63D

E680 0D 03 C3 40 C3 40 C3 8D : 66
E688 D5 C3 80 C3 12 04 E7 00 : D8
E690 E7 00 E7 C0 E7 C0 C3 C0 : B8
E698 C3 00 C3 80 C3 40 C3 00 : CC
E6A0 C3 80 C3 CB D5 EF 00 EF : 84
E6A8 00 EF 00 6F 00 2F 00 07 : 94
E6B0 40 03 40 01 40 01 00 FF : C4
E6B8 02 FF 01 FF 04 FF 00 FF : 03
E6C0 04 FF 00 55 0F 0D 0E 08 : 8A
E6C8 05 32 00 0F 01 0A 37 1E : A6
E6D0 00 00 00 00 00 00 00 50 : 50
E6D8 80 03 00 00 00 09 23 81 : 30
E6E0 08 24 85 09 34 06 42 36 : 6C
E6E8 91 61 01 49 34 91 61 09 : 6B
E6F0 72 11 0B 82 71 12 02 00 : 95
E6F8 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 25 01 82 B5 81 2B 3D 77 73BD

E700 CB FF 03 04 FF 05 22 22 : 19
E708 22 22 22 22 22 22 22 07 : F5
E710 00 07 40 07 40 00 0F 05 : A2
E718 FF 04 04 0F 00 00 00 00 : 16
E720 00 00 0F 35 0F 00 0F 40 : A2
E728 0F 40 00 00 FF 04 06 3F : 97
E730 00 00 00 00 40 40 07 47 : CE
E738 00 00 0F 40 0F 40 0F 00 : AD
E740 0F 00 00 0F 56 03 40 03 : BA
E748 40 07 00 07 00 03 40 00 : 91
E750 00 00 05 9B 03 40 07 40 : 2A
E758 1F 00 1F 00 00 00 00 06 : 44
E760 D5 FF 04 05 07 00 00 00 : E4
E768 00 4C 36 40 FF 04 16 00 : DB
E770 16 66 00 00 00 06 95 19 : 30
E778 18 00 00 00 34 67 0B E7 : A5
-----------------------------------
SUM: 6C 24 E5 A7 51 62 BB 3D 87ED

E780 00 00 2F 05 06 85 00 00 : BF
E738 00 17 37 00 2F 56 0F D6 : B8
E790 00 00 00 14 76 FF 04 05 : 92
E798 07 40 00 00 05 97 40 40 : 63
E7A0 00 00 00 00 FF 04 05 0F : 17
E7A8 00 00 00 6B 08 0B 08 40 : C6
E7B0 40 00 00 00 00 FF 04 07 : 4A
E7B8 03 00 00 00 14 56 15 48 : CA
E7C0 00 0F 37 00 FF 02 FF 01 : 47
E7C8 FF 04 FF 00 53 0C 0B 17 : 83
E7D0 04 04 1E 00 14 02 06 64 : A6
E7D8 64 00 00 19 19 07 05 0A : AC
E7E0 90 80 02 00 00 00 05 54 : 6B
E7E8 06 57 21 03 54 33 53 06 : 61
E7F0 32 04 35 61 21 52 81 21 : E1
E7F8 05 61 22 81 52 82 00 00 : DD
-----------------------------------
SUM: 7E AA 34 82 11 F3 67 BA DBBB

E800 B3 FF 03 06 FF 05 C1 C1 : 41
E808 C1 C1 C1 C1 C1 C1 C1 81 : C8
E810 C0 F7 00 81 C0 81 C0 99 : D2
E818 00 99 00 BB 00 BB 00 83 : 92
E820 80 83 80 DF 00 97 C0 97 : 50
E828 C0 81 40 99 40 93 40 C7 : F4
E830 40 CF 00 81 C0 85 AB 46 : C6
E838 0B 46 CD 00 C1 00 81 80 : E0
E840 C3 80 E1 C0 81 D4 26 9B : FA
E848 C0 99 40 D1 40 C3 80 F7 : E4
E850 0C D4 FF 04 14 81 00 00 : 78
E858 00 00 00 00 00 00 0E 06 : 14
E860 00 12 34 00 32 66 0E E6 : D2
E868 00 00 28 98 18 18 00 00 : F0
E870 2F 09 0F 09 00 00 EF 00 : 3F
E878 8D 40 81 6F 4A 0F 4A 8B : EB
-----------------------------------
SUM: 0A B1 5D A1 AA 56 69 8B B455

E880 40 8B 80 8F 00 8B 40 81 : 26
E888 C0 C1 80 E1 00 A1 C0 81 : C4
E890 F6 29 16 29 A1 C0 E3 00 : A2
E898 C7 CF 49 8F 80 89 40 81 : 38
E8A0 40 83 40 81 40 81 00 81 : C6
E8A8 00 81 D7 28 FF 02 FF 01 : 81
E8B0 FF 04 FF 81 3C 13 0F 1D : FE
E8B8 09 06 3C 00 0F 03 0B 37 : 9F
E8C0 28 00 00 12 12 0B 07 03 : 61
E8C8 99 99 08 00 00 00 08 44 : 86
E8D0 0E 32 46 07 53 04 48 06 : 32
E8D8 53 11 01 35 0B 12 31 43 : 2B
E8E0 34 02 57 13 31 23 35 09 : 32
E8E8 A1 52 21 12 0A 24 A1 52 : 47
E8F0 11 05 00 00 00 00 00 00 : 16
E8F8 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 0D 87 78 C5 56 76 9A 44 6C38
```

THE SENTINEL

●進化をたどる

構造化コンパイラSLANGと実数演算パッケージSOROBANを使用したシミュレーションプログラムです。ただし，シミュレーションとはいってもゲームとは違います。もっと純然とした意味でのシミュレーション，それも生物の進化を題材にしたプログラム，それがこのBUGSです。

このようなプログラムとしてはなんといってもライフゲームが有名ですね。もちろんS-OSでもライフゲームは作られています。S-OS用のライフゲームは1986年4月号に掲載されました。自由にカーソルコントロールもできない“MACE”用に作られたプログラムでしたので，直接ワークを書き換えるなど，かなり強引な方法を取らなければまともな表示を行えませんでした。それがいまや，さらに進んだシミュレーションがコンパイラで記述され，実数演算を行い，より多くの機種で動いています。いろいろと進化したものですね。

このプログラムは，人間にとってはただ眺めているだけの環境プログラムに等しいものです。なかなか利口にならないBUGたちをもどかしく思いつつも温かく見守ってあげましょう。なお，実行にはSLANG, SOROBAN, 実数演算ライブラリが必要です。

第84部
生物進化シミュレータ BUGS

●PC-9801で“SWORD”？

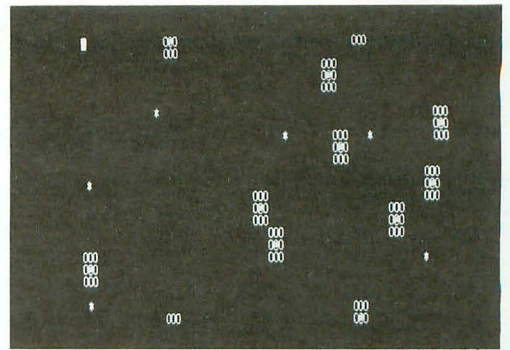
つ，ついに，PC-286, PC-9801用の“SWORD”が届いてしまいました（「PC-286およびPC-9801版」として投稿してくるというところがなかなか……）。

これは，かなり高速なエミュレータによるもので，ディスクは直接読み込み可能。仮想ドライブの設定も可。と，ざっと見ただけでもかなり気合の入ったプログラムになっています。加えて，X1版MAGICが無変更で走る？　となるともはやただ者ではありません。

BTRONマシンでS-OSを走らせるという野望が危うくなったいま（BTRONが），日本制覇にはPC-9801版は欠かせないものかもしれません。

●お知らせ

先月発表したCP/M用ファイルコンバータで機種により動作がおかしくなることが判明しました。現在調査中ですのでデバッグ情報はもうしばらくお待ちください。

また，以前から何度か投稿を呼びかけ，また要望も多かったS-OS用のオセロゲームは，掲載許可の際の制限が厳しくなっています。オセロゲームを作成中の方は原作のボード版のルールに忠実にするように心がけてください。

●S-OSの系譜（3）

S-OS“MACE”が発表された翌月の1985年7月号では，Z80のアセンブラZEDA（ゼータ）とデバッガZAID（ザイド）が発表されました。ZEDAはいわゆる「アブソリュートアセンブラ」で，ZEDAを使って作ったプログラムは特定のアドレスでしか動きません。しかし，エディタを内蔵しているためプログラム作成とマシン語への変換がZEDAを抜けることなく行えました。また，ZAIDはプログラムの開始番地がZEDAの直後となっており，この2つを組み合わせることでプログラムの作成からデバッグまでの一連の作業を簡単に切り替えることができたのです。

CP/Mなどではエディタを起動してソースプログラムを作り，次にエディタを終了してアセンブラを起動しマシン語に変換。そしてリンカを起動してリンク（アブソリュートアセンブラでない場合，リンクという作業をして初めて実行可能なプログラムができる）という作業は煩わしいものでした（この手順はX68000のHuman68k上でマシン語プログラムを作る場合も同様です）。大きなプログラムは作れないというデメリットはあったものの，ZEDAのこの方式は多くの読者に支持されました。

S-OSのメリットは，異なる機種で同じマシン語プログラムを実行できることだけではありません。掲載されるプログラムはすべてソースリスト付きで発表されました。S-OSのアプリケーションは言語関係のものが多く，「プログラミング言語の仕組み」を理解するにはもってこいだといえるでしょう。マシン語の格好の教材が提供されているわけです。

続く8月号では，マシン語からアセンブリ言語のソースプログラムを生成するZING（ジング）が発表され，それまでソースプログラムを入力するのが面倒だった方でも，ダンプリストを入力しZINGでソースプログラムを生成することが可能となりました。これでマシン語プログラミングの三種の神器（アセンブラ，デバッガ，ソースコードジェネレータ）が出揃い，S-OSアプリケーションの投稿が始まります。

全機種共通S-OS"SWORD"要

生物進化シミュレーションBUGS

SLANGと実数演算パッケージSOROBANを使った生物進化シミュレーションプログラムです。画面上で自然淘汰モデルを再現してみましょう。そのほか，ローダの変更がありますので，必ず目を通しておいてください。

Nishimura Susumu
西村 進

BUG，とはいってもあのプログラムの天敵「バグ」のことではありません。じゃあいったいなんなのか。実はこれは，日経サイエンス(日経サイエンス社)1989年7月号「コンピュータレクリエーション」で紹介されている生物進化シミュレーションをSLANGでプログラムしたものなのです。

ここでは進化とはいっても，サルがヒトになったりするわけではなく，水中に棲むアメーバが食物となるバクテリアを探し求めて動き回るうちに，その探索パターンがより効率的なものへと進化していくというものです。

進化ってナンダ?

「進化」といわれて，誰もがまず思い出すのがダーウィンの『種の起原』ではないでしょうか(ほんとかね?)。すべてはこの本から始まったのです。

ダーウィンが唱えたのは「自然選択説」というものです。すなわち，生物はみんな多産であるために生存競争が起こり，生まれつき生活に有利な変異を持つものが生き残って子を残し（いわゆる適者生存というやつです)，その子にこの変異が伝えられるので，生物はその方向に次第に進化していくであろうというものです。

もっとも話はこれほど単純ではなく，進化は突然変異によるものであるとする「突然変異説」という学説もありますし，ほかにも隔離，倍数体や雑種の形成など，いろいろな要素がからんできます。しかし，突然変異を素材として自然選択が働き，それが進化の原因をなす，と考えるのが一般的です。

それではいったいなにが，親から子へと変異を伝えていくのでしょうか。そうです，いわゆる遺伝子というものです。生化学的にいうとDNA（デオキシリボ核酸）ということになります。

いわゆる生物というものは，皆このDNAというものを持っています。そして，このDNAというものはその親から伝えられたものなのです。ですから子が親に似るのは当然というわけです。しかし，子が親とまったく同じDNAを受け継ぐとは限りません。なぜなら突然変異の可能性があるからです。

つまり，親のDNAにちょっと変異が起こったものが子のDNAになるというわけです。

誤解がないようにいっておきますと，人間の子供が親と違うところを持つのは，別に突然変異によるわけではありません（もちろん突然変異によるところも少しはあるのでしょうが)。なぜならひと口に親といっても父親と母親がいて，2人の遺伝子を受け継ぐからです。

仮にここで，父親がXX,母親がYYという遺伝子を持っていたとすると子供はXとYの中間であるXYという遺伝子を受け継ぐことになるわけです。ですから子供は父親，母親両方ともに対して，似ているところ，似ていないところを持つわけです。

さらになぜ同じ両親から生まれたのに容姿の違う兄弟が生まれるか，またどうして男の子が生まれたり女の子が生まれたりするのかということですが，父親の遺伝子がXx，母親の遺伝子がYyであるとすると，生まれた子供の遺伝子はXY/Xy/xY/xyという4通りの可能性があるのがわかりますよね。

このほかにも優性/劣性遺伝子，遺伝子の組み換えなど，さらに多くのことがからんできます。だいたい人間ともなると実効性のある遺伝情報の量たるや，すさまじいものがありますから，とても説明しきれるものではありません。そこまでいくと完全に生物の授業になってしまいますから（メンデルの遺伝の法則，って覚えていますか)，興味のある方は自分で調べてみてください。

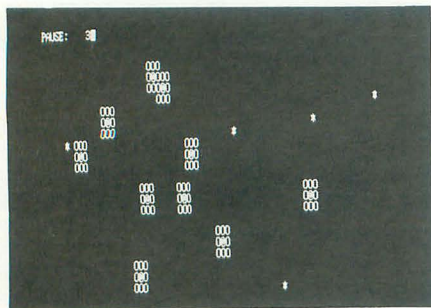
とまあ，つきつめていくとこの話はかなり面倒くさいことになってしまいますが，ここで扱うのはもっとも単純なモデルです。すなわち，あるアメーバからさらに2つの同様のアメーバが生まれてくるとします。この場合，アメーバは単体で子を生むことになりますから，人間と違って（人間は男性と女性による有性生殖)，無性生殖ということになります。当然，子のDNAは親のDNAとほぼ同じものになります。ほぼ，といったのは突然変異というものがあるからです。

こうして新しいアメーバが誕生してくるわけですが，エサ（バクテリア）の数というのはたかが知れていますから，当然アメーバの数が過多になり，生存競争が起こるわけです。

その結果としてより効率的にエサをとることのできるアメーバ（というよりも，より効率的にエサをとることのできるような遺伝情報をDNAとして持っているアメーバ)だけが生き残る，すなわち「適者生存」となるわけです。これがここでいうところの「進化」なのです。

BUGの世界にて

BUGの世界の規則は前述の別冊サイエンスに詳しいのですが，ここでも重要な点について説明しておきましょう。BUGは仮想の生物ですが，生物であるからには1匹1匹みな固有のDNAを持っています。

そして，そのBUGの行動は自己のDNAに入っている情報をもとに決定されます。すなわち，BUGが1ステップごとにどちらの方向に，どのぐらい回りやすいかを決定す

るわけです。

具体的には，DNAは回転する方向(６方向）ごとに回転しやすさを表す数F/R/HR/RV/HL/L（それぞれ前方/右前方/右後方/後方/左後方/左前方を表す）を情報として持っています。そしてこの情報により，回転する方向が決定されるわけです。

たとえばBUGが右前方を向く確率は，

$$2^R/(2^F+2^R+2^{HR}+2^{RV}+2^{HL}+2^L)$$

となります。

このようにして各々のBUGはDNAの情報と乱数により１ステップごとに回転方向を決定し，その方向へ１歩進みます。そして進んだ場所にエサであるバクテリアがいればそれを食べることができるわけです。

エサのバクテリアはある規則に従って降ってきますが，このバクテリアを１匹食べるごとにBUGは自分のエネルギーを一定値だけ増やすことができます。しかしBUGは１ステップ動くごとにエネルギーを１ずつ消費してしまいますから，バクテリアにうまくありつけないBUGは死んで消えてしまいます。

逆に，バクテリアの捕り方がうまいBUGは，体内にエネルギーをどんどん蓄えていき，十分に成熟した（ある回数以上動いた）時点で分裂して２匹の新しいBUGを生み出します。その際にはエネルギーは３匹（親１匹，子２匹）で３等分し，子のDNA情報は親のDNA情報の値から適当な数を増減したものとなります。ただし一方の子供BUGのDNA情報の値を増したら，もう一方の子供BUGのDNA情報は同じ値だけ減らすものとします。

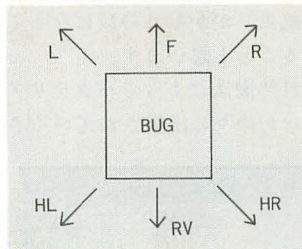
このような手続きを１ステップずつ行うと，バクテリアの探索パターン（すなわちDNA情報）の悪いBUGは自然淘汰され，良い探索パターンを持つBUGが繁殖するようになります。これこそが，「自然選択」の結果なのです。

ちなみに，バクテリアの降ってくる確率が画面上のどこでもだいたい等しければ，「巡航虫」というあまり向きを転回しない（Fの値が大きい）BUGが繁栄しますが，局地的に確率が高い場所があれば，そこでは「グルグル回り」という一方向にだけ回る確率の高いBUGが繁栄します。

天地創造

このシミュレーションではいくつかの定数や関数を変えることによって環境（BUGの住んでいる世界）の設定を変更することができます。

図1　BUGの動き方

図2　親BUGと子BUG

まず，定数です。KETAとMAXLはそれぞれ画面の幅と行数を表しています。ですからwidth(KETA)で，表示範囲は(0,0)－(KETA－2,MAXL－1）ということになります。40桁モード固定のユーザーの方はKETA＝40としておいてください。一般には表示範囲はできるだけ広くしておいたほうがよい結果が得られると思います。

また，BUGMAXはBUGの最大数，FIRSTBUGは最初のBUGの数で，BUGLIFEはBUGの寿命，REPYEARとREPENGYはそれぞれ繁殖可能となる年齢とエネルギーです。

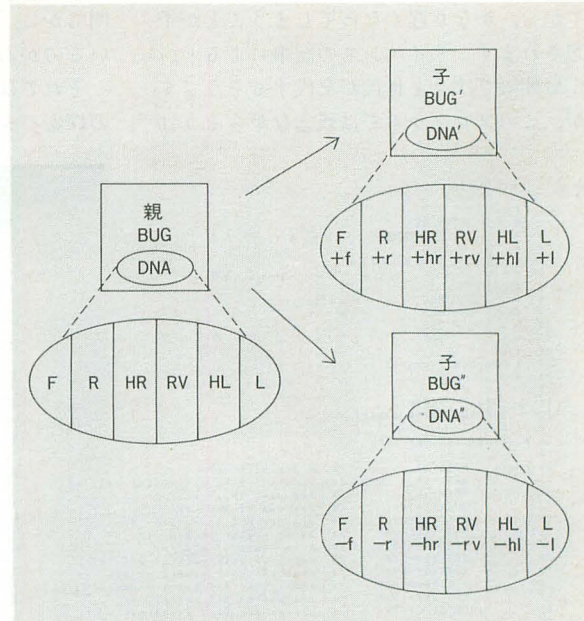
さらに，FIRSTENGYは最初のBUGが持っているエネルギー，ENGYMAXは１匹のBUGが持つことのできる最大のエネルギー，FEEDENGYはバクテリアを１匹食べることによって得られるエネルギーを表しています。

次に関数ですが，amedas (x,y) という関数を変えることができます。これは１ステップごとに画面上のすべての点で呼び出され，バクテリアが降ってくるかどうかを値として返します。すなわち，降ってくればTRUEを，降ってこなければFALSEを返します。

これを改造することによって，先ほど述べたように局地的にバクテリアの豪雨が降るようにしてみても面白いでしょう。

入力&実行

このプログラムを実行するにあたってはいうまでもなくS-OS“SWORD”が必要ですが，さらに，SLANG，浮動小数点パッケージSOROBAN同ライブラリSOROBAN.LIBが必要です。

ここでひとつやってもらいたいことがあります。ニューローダについてなのですが，このプログラムは#MEMAXから下へ64バイトをワークとして確保しています。しかし，機種によってはこれをもう１バイト確保しないとうまく動作しないことがありえます。そこでニューローダの300B_Hを40_Hから41_Hへと書き換えてください。

さて，本題に戻って入力ですが，まずBUGS.SLをエディタから入力します。このとき，テープユーザーの方はファイルのインクルードができませんので，インクルードしている位置に前もってSOROBAN.LIBを付け加えておいてください。

そして次に，まずローダでSOROBANを適当な位置にロードしてから，SLANGでコンパイルし，ファイル名BUGSでセーブしてください。

この際にSOROBANが破壊されてしまった場合には，再ロード後，実行してください。BUGがバクテリアを求めて動き出します。＊がバクテリアで，３×３キャラクタの四角形のものがBUGです。

最後に

高速性に関してはそれなりに気をつかったつもりなのですが，少々遅くてイライラするので（必ずしも実数演算のせいではない），スキップ機能をつけました。実行中にSを入力すると，10ステップ分は画面に出力せず，計算のみ行います（何ステップ分だけスキップするかは定数SKIPによっている)。

また，スペースを入力すると一時停止して画面左上に現在のBUGの数を表示します。なお，ブレイクで中断です。

初めはグラフィック画面で表示する予定

だったのですが，文字画面でこの状態ですから，かなり遅くなってしまうことが予想されます。サイエンスの記事によると，1分間隔で次々と世代が交代するそうですが，このプログラムでは残念ながらそうはいきません。しかし，1時間半も待てば，明らかに「巡航虫」がかなりの数になっているのがわかることと思います。

それでは，自然淘汰によってバクテリアの探索パターンが徐々に洗練されていくさまを「ごゆっくりと」お楽しみください。

Profile

◇西村さんは京都府にお住まいの19歳，現在大学2年生です。マイコン歴は約4.5年でMZ-2500を使用しています。このコーナーにはまた近いうちに登場していただくことになると思います。

リスト1　BUGS.SL

```
  1 /*
  2  *        BUGS.SL
  3  *            原生動物進化シミュレーション
  4  *
  5  *            サイエンス7月号
  6  *                コンピューターレクリエーションより
  7  *
  8  *           BUG          Feed(Bacteria)
  9  *           000            *   *
 10  *           0@0             *    *
 11  *           000          *    *    *
 12  *
 13  *  Do BUGs survive or die off?
 14  */
 15
 16     ORG    $3000;
 17     OFFSET $9000-$3000;
 18
 19 #include SOROBAN.LIB
 20
 21 const   KETA-80,          /*  width KETA  */
 22         MAXL-24,          /*  locate(?,MAXL) is prohibited  */
 23         SKIP-10,          /*  how many skip  */
 24         BUGMAX-100,       /*  maximum of BUG  */
 25         BUGSIZE-32,       /*  size of BUG structure  */
 26         FIRSTBUG-10,      /*  first number of BUG  */
 27         BUGLIFE-5000,     /*  life limit of BUG  */
 28         REPYEAR-100,      /*  reproductable year of BUG  */
 29         REPENGY-500,      /*  reproductable energy of BUG  */
 30         FIRSTENGY-40,     /*  particular energy of first BUG  */
 31         ENGYMAX-1500,     /*  maximum of BUG energy  */
 32         FEEDENGY-40;      /*  energy par Feed(Bacteria) */
 33
 34 array byte vramr[KETA-2][MAXL-1],byte svramr[KETA-2][MAXL-1];
 35 var   byte vram[][MAXL-1],byte svram[][MAXL-1];
 36 array byte bugdata[BUGMAX*BUGSIZE-1];    /*  BUGデータ領域  */
 37 array word bugptr[BUGMAX-1];      /*  BUGへのポインタ  */
 38 array byte geta[@sng],        /*  確率を実数から整数に変換する倍率(65535)  */
 39       byte zero[@sng],        /*  (float)0  */
 40       byte zwei[@sng];        /*  (float)0  */
 41 array word xmove[5] - [ %0, %2, %2, %0,%-2,%-2 ],
 42       word ymove[5] - [ %2, %1,%-1,%-2,%-1, %1 ];
 43
 44 var bugnum;            /*  BUGの数  */
 45
 46 main()
 47 var sk;
 48     [
 49     vram-vramr;        /*  vramは配列vramr[][]へのポインタ  */
 50     svram-svramr;      /*  svramは配列svramr[][]へのポインタ  */
 51     width(KETA);
 52     @single();         /*  計算は単精度  */
 53     setfcon();         /*  実数定数の定義  */
 54     initbug();         /*  BUGをFIRSTBUG匹発生  */
 55     vraminit();        /*  仮想VRAM初期化  */
 56 //
 57     print("¥c");
 58     sk-0;
 59     while(bugnum!-0)
 60         [
 61         if (newgen()!-TRUE)
 62             [
 63             locate(0,0);
 64             print("Too many BUGs were reproducted.¥n");
 65             exit;
 66             ]
 67         case(sk)
 68             [
 69             0:  display();             /*  表示  */
 70             1:  [
 71                 rewrite();             /*  全面表示  */
 72                 sk--;
 73                 ]
 74             others: sk--;
 75             ]
 76         swap(&vram,&svram);
 77         case(inkey(0))
 78             [
 79             ' ':        pause();    /*  PAUSE & PRINT bugnum  */
 80             's','S':[               /*  SKIP mode  */
 81                     sk-sk+SKIP;
 82                     locate(0,0);
 83                     print("SKIP");
 84                     ]
 85             $1B:    [
 86                     locate(0,0);
 87                     exit;
 88                     ]
 89             ]
 90         ]
 91     if (bugnum--0)
 92         print("¥cBUGs died off.¥n");
 93     ]
 94
 95 setfcon()
 96     [
 97     @cvutf(geta,65535);            /*  geta-65535  */
 98     @cvutf(zero,0);                /*  zero-0  */
 99     @cvutf(zwei,2);                /*  zwei-2  */
100     ]
101
102 initbug()
103     var word p[];
104     var i;
105     [
106     for (i-0; i<BUGMAX; i++)
107     bugptr[i]-bugdata+BUGSIZE*i;
108     for(bugnum-0; bugnum<FIRSTBUG; bugnum++)
109         [
110         p-bugptr[bugnum];
111         p[0]-1;               /*  1 year old  */
112         p[1]-FIRSTENGY;       /*  energy-Default  */
113         p[2]-rnd(KETA-1);     /*  BUG's x  */
114         p[3]-rnd(MAXL);       /*  BUG's y  */
115         p[4]-rnd(6);          /*  random turn  */
116         for(i-5; i<11; i++)
117             p[i]-rnd(7)-3;    /*  random gene  */
118         setprb(&p[5],&p[11]);   /*  set turn distribution  */
119         ]
120     ]
121
122 vraminit()
123     var x,y;
124     [
125     for(y-0; y<MAXL; y++)
126         for(x-0; x<KETA-1; x++)
127             vram[x][y]-(amedas(x,y))?'*':' ';
128     ]
129
130 svraminit()
131     var x,y;
132     [
133     for (y-0; y<MAXL; y++)              /*  Sub-VRAM 初期化  */
134         for (x-0; x<KETA-1; x++)
135             svram[x][y]-(amedas(x,y) or vram[x][y]--'*')?'*':' ';
136     ]
137
138 sort(n)      /*  インチキソート(ガーベジコレクション)  */
139     var i,j,temp;
140     [
141     for (i-0; i<n; i++)
142         if (memw[bugptr[i]]--0)
143             [
144             temp-bugptr[i];
145             for (j-i; j<n-1; j++)
146                 bugptr[j]-bugptr[j+1];
147             bugptr[n-1]-temp;
148             ]
149     ]
150
151 newgen()     /*  BUGSが異常増殖すればFALSE、そうでなければTRUE  */
152     var death;
153     [
154     svraminit();
155     death-grow();          /*  One step forward  */
156     if (death!-0)
157         sort(bugnum);   /*  Garbage collection  */
158     if ((bugnum-bugnum-death)--0)          /*  BUG die off  */
159         return(TRUE);
160     return(newborn());  /*  Make new BUG  */
161     ]
162
163 grow()
164     var i,turn,death;
165     var word p[];
166     [
167     for (i-0,death-0; i<bugnum; i++)
168         [
169         p-bugptr[i];                /*  p-&(i番目のbugdata)  */
170         turn-getturn(&p[11]);    /*  prb[]に従ってturnを返す  */
171         p[4]-turn-(p[4]+turn) mod 6;                   /*  turn-newturn
*/
172         p[2]-(p[2]+xmove[turn]+KETA-1) mod (KETA-1);  /*  x-x+xmove(turn)
*/
173         p[3]-(p[3]+ymove[turn]+MAXL) mod MAXL;        /*  y-y+ymove(turn)
*/
174         p[1]-p[1]+putbug(p[2],p[3])*FEEDENGY-1; /*  put bug and eat virus
*/
175         if (p[1]>ENGYMAX)
176             p[1]-ENGYMAX;              /*  energy<-ENGYMAX  */
177         p[0]++;                        /*  year++  */
178         //
179         if (p[0]>BUGLIFE or p[1]--0)    /*  寿命or餓死  */
180             [
181             p[0]-0;      /*  year-0  */
182             death++;
183             ]
184         ]
185     return(death);
186     ]
187
188 newborn()        /*  Memory Over したときはFALSE、そうでないときはTRUE  */
189     var i,j,birth,delta;
190     var word p[],word pn1[],word pn2[];
191     [
192     for (i-0,birth-0; i<bugnum; i++)
193         [
194         p-bugptr[i];
195         if (p[0]>REPYEAR AND p[1]>-REPENGY)      /*  分裂可能  */
196             [
197             if (bugnum+birth+2>-BUGMAX)          /*  Memory Over  */
198                 return(FALSE);
199             //
200             pn1-bugptr[bugnum+birth];
201             pn2-bugptr[bugnum+birth+1];
202             pn1[0]-pn2[0]-1;                     /*  1 year old  */
203             p[1]-pn1[1]-pn2[1]-p[1]/3;           /*  divide Energy equally  */
204             pn1[2]-pn2[2]-p[2];                  /*  x  */
205             pn1[3]-pn2[3]-p[3];                  /*  y  */
206             pn1[4]-pn2[4]-p[4];                  /*  turn  */
207             //
208             for (j-5; j<11; j++)         /*  recieve DNA  */
209                 [
210                 delta-rnd(7)-3;
211                 pn1[j]-p[j]+delta;
212                 pn2[j]-p[j]-delta;
213                 ]
214             setprb(&pn1[5],&pn1[11]);
215             setprb(&pn2[5],&pn2[11]);    /*  set newborn's turn distributio
n  */
216             birth-birth+2;
217             ]
218         ]
219     bugnum-bugnum+birth;
220     return(TRUE);
221     ]
222
223 setprb(word gene[],word prb[])
224     var i,¶p;
225     array byte gsum[@sng],byte gpow[6][@sng],byte dummy[@sng];
226     [
227     @move(gsum,zero);          /*  *gsum-0  */
228     @move(gpow[0],zero);       /*  *gpow[0]-0  */
229     for (i-1; i<7; i++)
230         [
231         fp-@cvitf(gpow[i],gene[i-1]);      /*  *fp-*gpow[i]-(float)gene[i-1]
*/
232         @add(gsum,gsum,@pow(fp,zwei,fp));/*  *gsum-*gsum+pow(2,*fp)  */
233         ]                                  /*  *fp-pow(2,*fp)  */
234     fp-gpow[0];                    /*  *fp-0  */
235     for(i-0; i<6; i++)
236         [
237         @add(fp,fp,gpow[i+1]);  /*  *fp-*fp+*gpow[i+1]  */
238         @div(dummy,fp,gsum);      /*  *dummy-*fp/*gsum  */
239         prb[i]-@cvftu(@mul(dummy,dummy,geta));
240                                 /*  prb[i]-(unsigned)(*dummy**geta)  */
241         ]
242     ]
```

```
243
244 getturn(word prb[])        /*  prbに従ってturnを1つ返す。  */
245     var i,ri;
246     [
247     ri=rnd(65535);
248     for (i=0; i<5; i++)
249         if (ri<prb[i])
250             return(i);
251     return(5);
252     ]
253
254 putbug(x,y)       /*  (x,y)を左肩の座標としてSub-VRAMにBUGを書き込む  */
255     var feed,     /*  食べたウイルスの数を返す  */
256         dx,dy,px,py;
257     [
258     for (feed=0,dy=0; dy<3; dy++)
259         for (dx=0; dx<3; dx++)
260             case(svram[px=((x+dx) mod (KETA-1))][py=((y+dy) mod MAXL)])
261                 [
262                 '*':    [
263                         feed++;
264                         svram[px][py]=(dx==1 and dy==1)?'@':'O';
265                         ]
266                 ' ':    svram[px][py]=(dx==1 and dy==1)?'@':'O';
267                 ]
268     return(feed);
269     ]
270
271 display()
272     var x,y,c;
273     [
274     for (y=0; y<MAXL; y++)
275         for (x=0; x<KETA-1; x++)
276             if ((c=svram[x][y])!=vram[x][y])
277                 [
278                 locate(x,y);
279                 print(str$(c,1));
280                 ]
281     ]
282
283 rewrite()
284     var x,y;
285     [
286     for (y=0; y<MAXL; y++)
287         for (x=0; x<KETA-1; x++)
288             [
289             locate(x,y);
290             print(str$(svram[x][y],1));
291             ]
292     ]
293
294 swap(word p[],word q[])
295     var temp;
296     [
297     temp=p[0];
298     p[0]=q[0];
299     q[0]=temp;
300     ]
301
302 pause()
303     var x;
304     [
305     locate(0,0);
306     print("PAUSE:",deci$(bugnum));
307     while(inkey(2)==' ')
308         ;
309     for (x=0; x<11; x++)
310         [
311         locate(x,0);
312         print(str$(vram[x][0],1));
313         ]
314    ]
315
316 amedas(x,y)
317     [
318     if (x<40)
319         return(rnd(20000)==0);
320     else
321         return(rnd(20000)<4);
322     ]
323
```

全機種共通システムインデックス

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

QUESTION and ANSWER

初めてお便りします。先日SWORDで入力途中のファイルをロードしようとして，誤ってファイルをKILLしてしまいました。バックアップも取っていなかったので，このままだと長い時間をかけて入力したプログラムが無駄になってしまいます。削除してしまったファイルを復活させることはできないのでしょうか。編集室の皆さん，どうか助けてください。使用機種はX1です。

徳島県　岡田　栄子

この手の質問にはディレクトリに記録されている内容さえ知っていれば対処することができます。

X1turboおよびX1で使われているディスクBASICやSWORDはレコード0をIPL，レコード14をFAT，レコード16～31をディレクトリ領域，レコード32～1279をデータ領域（2Dの場合）に使用しています。このなかでファイルネームや最後にファイルを保存した時刻，パスワードなどの情報を記録している部分がディレクトリです。

ひとつのファイルは32バイトの情報を持っていて，1レコードには8つのファイル情報が記録されます。ここにはファイルの格納アドレスやファイルサイズなどの情報も当然記録されています。

S-OSでは，ロードコマンドのLと，削除コマンドのKがキーボード上で隣同士の位置にあるために，うっかりしているとロードしようとしたファイルを消しちゃった，なんてことも起こります。

万一ファイルを消してしまっても，そのあとそのディスクに新しいファイルを書き込んだり既存のファイルを更新したりしなければ，そのファイルの内容はちゃんと残っています。しかし4Kバイトを超えるファイルについては復活が難しくなってきます。というのは，ディスクBASICでは4Kバイトを1クラスタという単位で扱っていて，このクラスタという概念のもとにディスクへの記録が行われているからです。

ファイルが最初にどこのクラスタから始まるかはディレクトリに記録されています。ところが4Kバイトを超えると複数のクラスタにまたがってデータが記録されることになるので，どのクラスタからどのクラスタにデータがつながっているのかを把握する必要が出てきます。そのつながりを記録しているのがFATというところです。

ファイルを削除命令で消してしまうと，そのファイルの属性（バイナリだとかASCIIだとかの）を示す部分がクリアされるだけでなく，FATのつながりまでもクリアされてしまうのです。ですから4Kバイトを超えるファイルをどうしても復活させたいときには，手探りでFATのつながりを探し出すことになります。これは暗い洞窟の中に明りもつけずに入っていくようなもので，相当の経験とカンのよさが要求されます。初心者の方にはまず無理でしょう。

そこで，とり急ぎファイルの格納アドレスを表示して，なおかつ削除されたファイルがあれば，それを復活させるようなプログラムを作ってみました（リスト1-A)。

非常に汚いプログラムで恥ずかしいのですが，そこらへんは笑って許してください。実行するといろいろ質問されますのでキーボードから適当な数字を入力してください。あとは勝手にファイルの格納アドレスを表示してくれます。

ちなみにASCIIファイルやBASICファイルではファイルサイズのみが表示されます。また，途中に削除されたファイルがあると

リスト1

A)
```
100 '************************
110 '  DISK RESCUE PROGRAM
120 '************************
130 '
140 PRINT "Input drive No. (0 or 1) ";
150 DR$=INPUT$(1):IF (DR$<>"0" AND DR$<>"1") THEN 150
160 PRINT DR$
170 PRINT "Search file mode (1-Bin 2-Bas 3-Asc) ";
180 M$=INPUT$(1):IF M$<"1" OR M$>"3" THEN 180
190 M=VAL(M$):IF M=3 THEN M=4
200 PRINT M$
210 '
220 ' Read directory
230 '
240 FOR REC=16 TO 31
250 DEVI$ DR$+":",REC,A$,B$
260 GOSUB "DISPLAY"
270 AA$=A$:A$=B$
280 GOSUB "DISPLAY"
290 B$=A$:A$=AA$
300 DEVO$ DR$+":",REC,A$,B$
310 NEXT
320 END
330 '
340 ' Display directory
350 '
360 LABEL "DISPLAY"
370 FOR J=1 TO 128 STEP 32
380 X=ASC(MID$(A$,J,1))
390 IF X=255 THEN 490
400 A=VAL("&H"+HEX$(ASC(MID$(A$,J+21,2))))*256
410 A=A+VAL("&H"+HEX$(ASC(MID$(A$,J+20,2))))
420 B=VAL("&H"+HEX$(ASC(MID$(A$,J+19,2))))*256
430 B=B+VAL("&H"+HEX$(ASC(MID$(A$,J+18,2))))
440 C=VAL("&H"+HEX$(ASC(MID$(A$,J+23,2))))*256
450 C=C+VAL("&H"+MID$(A$,J+22,2))
460 IF (X AND M)<>0 GOSUB "FILENAME":GOSUB "ADRS"
470 IF X=0 THEN GOSUB "FUKKATU"
475 ' 470ｷﾞｮｳ ｦ REM ﾆ ｽﾙﾄ KILL FILE ｦ ﾑｼ ｽﾙﾖｳﾆ ﾅﾙ
480 NEXT
490 RETURN
500 '
510 LABEL "FILENAME"
520   KMODE1 'TURBO USER ﾊ ' ｦ ﾄﾙ
530 PRINT:PRINT MID$(A$,J+1,13);".";MID$(A$,J+14,3);" : ";
540   KMODE0 'TURBO USER ﾊ ' ｦ ﾄﾙ
550 RETURN
560 '
570 LABEL "ADRS"
580 IF M=1 THEN PRINT HEX$(A);"H-";
590 PRINT HEX$(A+B-1);"H";
600 IF M=1 THEN PRINT CHR$(9);"Exec adrs : ";HEX$(C);"H";
610 RETURN
620 '
630 LABEL "FUKKATU"
640 GOSUB "FILENAME":GOSUB "ADRS"
650 PRINT " ﾌｯｶﾂ ｻｾﾏｽｶ (Y-0 N-1) ";
660 D$=INPUT$(1):IF D$<"0" OR D$>"1" THEN 660
670 PRINT D$:IF D$="1" THEN 830
680 PRINT "File mode (1-Bin 2-Bas 3-Asc) ";
690 N$=INPUT$(1):IF N$<"1" OR N$>"3" THEN 690
700 IF N$="3" THEN N$="4"
710 MID$(A$,J,1)=CHR$(ASC(N$)-48)
720 DEVI$ DR$+":",14,Y$,Z$
730 MEM$(&HC000,128)=Y$
740 MEM$(&HC080,128)=Z$
750 ADRS=&HC000+VAL("&H"+HEX$(ASC(MID$(A$,J+30,2))))
760 SI=VAL("&H"+HEX$(ASC(MID$(A$,J+19,2))-1))
770 IF SI>&HF THEN SI=&HF
780 POKE ADRS,&H80+SI
790 Y$=MEM$(&HC000,128)
800 Z$=MEM$(&HC080,128)
810 DEVO$ DR$+":",14,Y$,Z$
830 RETURN
```

B)
```
100 '************************
110 '  FAT BACKUP PROGRAM
120 '************************
130 '
140 ' FAT ｦ ﾎｿﾞﾝ ｽﾙ
150 '
170 DR$="1:"  ' DRIVE No.
180 DEVI$ DR$,14,A$,B$
190 MEM$(&HC000,128)=A$
200 MEM$(&HC080,128)=B$
210 SAVEM DR$+"Fat backup.Obj",&HC000,&HC0FF
220 END
230 '
300 ' ﾎｿﾞﾝ ｼﾀ FAT ｦ ﾓﾄﾞｽ
305 '
310 DR$="1:"  ' DRIVE No.
320 CLEAR &HC000
330 LOADM DR$+"Fat backup.Obj"
340 A$=MEM$(&HC000,128)
350 B$=MEM$(&HC080,128)
360 DEVO$ DR$,14,A$,B$
370 END
```

復活させるかどうかを聞いてきます。そのときにそのファイルがバイナリファイルかASCIIファイルかBASICファイルかを尋ねてきますが，もし忘れてしまっていたらひととおり選んでやってみてください。そのファイルが4Kバイト未満で，なおかつKILLした直後の状態のディスクなら100%復活します。

4Kバイト以上のBASICファイルでも途中までは復活しますが，そのようなファイルを読み込むと暴走することがあります。うまくロードできない場合には，残念ながら復活不可能ですのであきらめてください。なお，このプログラムはディスクに対して書き込みを行うので，デバッグが完了するまではバックアップディスクに対して実行するようにしてください。

FATを復活させる作業はかなり大変な作業なので，おまけにFATのバックアップを取っておくプログラムを作ってみました(リスト1-B)。Strat up.Basなどに加えておけば便利でしょう。

日頃からディスクのバックアップを取っておけば，このような事態になって慌てることはありません。できるだけまめにバックアップを取るようにしておきたいものです。　　　　　　　　　(影山　裕昭)

OS-9/X68000を使っています。この前OS-9のディスクを抜いてほかのディスクを入れてディレクトリを見ようと思ったらエラーが出てしまいました。おかしいと思っていろいろなコマンドをやってみたのですがうんともすんともいいません。どうなってるんですか？

佐賀県　石田　俊彦

OS-9はHumanと違ってコマンドに内部コマンドというものがないからです，ではいきなり答が2行で終わってしまうのでもう少し詳しく説明することにしましょう。まずHuman68kでコマンドラインから指定されたコマンドを実行する場合，まずそれが内部コマンドであるかどうかを見にいき，そのあとでカレントディレクトリ，PATHコマンドで指定されたディレクトリにそのコマンドがあるかどうかを探し……という具合になっています。

内部コマンドというのはあらかじめHumanの内部（つまりメモリ上に）に設定されたコマンドでDIR，REN，COPYなどがこれにあたります。ですから，HumanではDIRはシステムディスクがどこにも入ってなくても使えるわけです。

ところが，OS-9はそうではありません。コマンドをキーボードから受け取るシェル(Humanのcommand.xに相当します）は自分自身ではコマンドを持っていないで，ほとんどのコマンドをディスク上の/D0/CMDSかH0/CMDSというディレクトリ（あるいはユーザーによってCHXコマンドで設定されたディレクトリ）に探しにいきます。DIRも例外ではありません。そのため，システムディスクが入ってなければシェルはうんともすんともいいません。

で，まず第一の解決策ですがフォーマットして作ったすべてのディスクにCMDSをコピーしてやってやるという手があります。これならばどのディスクを使ってもコマンドが動かないということはありません。しかし，コマンドを自作するのが好きな人の場合，使いたいコマンドがそのディスクに入っていないということもあり得ます。

それに，OS-9のディスクアクセスはお世辞にも速いとはいえないような速度ですので，フロッピーベースでOS-9を普通に動かそうとすればコマンド実行のたびにフロッピーをがりがりとアクセスしにいき，はっきりいって使いものになりません（フロッピーのような容量も小さく遅い補助記憶装置をこのテのOSで使うほうが非常識なのかもしれませんが)。

そこで第2の方法として，OS-9にもちゃんと抜け道（というほどおおげさなものではないが）を利用する手があります。その名をメモリモジュールといいます。これはOS-9のシェルはコマンドを実行するときメモリのなかにそのコマンドを残しておき，あとでそのコマンドが呼ばれたときにディスクを読みにいかなくてもいいようにしています（呼ばれたコマンドがスティッキィモジュールである場合)。

MDIRで確認してみればわかると思いますが，いままで使用したコマンドがほとんどすべてメモリ内に入っているはずです。それならどうにかしてそのメモリ中にコマンドをもぐり込ませてしまえばHumanでいう内部コマンドの代わりになるわけです。

それを行うのがLOADコマンドです。これで可能な限りのコマンドをメモリに詰め込めばそこそこ使えるようになるでしょう。また「MFREE　-e」でセグメント単位での空きエリアを見ることができるので，どのくらいのコマンドが入れられるかわかります。また，メモリが足りなくなった場合はUNLINKコマンドでメモリから削除することができます。

ま，それで使い心地は多少はよくなるかもしれませんがはっきりいって気休めぐらいにしかなりません。OS-9のディレクトリの構造は複雑にできているうえにOS-9自体が余りに細かくモジュール化されており，ディスクひとつ読むのにいろいろなところを通るためディスクのリードは遅くなってしまいます。また，これをどうにかして速くしようとしてもOS-9の構造上，ディスクのベリファイを省かせるぐらいしか手がありません。

ですからディスクの読み書きが絶対的に遅いフロッピーディスクではOS-9でのオペレーションはかなりきついものがあります。OS-9をまともに使いたいのであればハードディスクは必須のものですのでぜひともハードディスクを買ってください。特にOS-9ではハードディスクは初期投資にみあうだけの絶大な性能強化が見込めます。いまやハードディスクやメモリもさほど高くないのですし，20Mのハードディスクと1Mバイトの増設RAMくらいの装備は，思いきってつけてみてはどうでしょうか。

(古村　聡)

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。毎日暑いですね。各地で催される夏祭りもたけなわといったところでしょうか。皆さん，寝冷えと夏バテにはご注意を。

一般

▶NIESコンピュータ事情(1)台湾
最近コンピュータ業界でも追い上げの目立つアジアNIESの各国についてレポート。第1回は台湾パソコン界の実情をコンピュータショウを中心に紹介。——編集部，ASAHIパソコン，18号，96-101pp.

▶特集C言語入門
C言語入門の手引き，主なC言語の特徴や使いこなし方など。——編集部，I/O，8月号，90-127pp.

▶パソコンCPUの主流80386/80486入門
CPUの発達の歴史から，最新プロセッサのアーキテクチャまでを解説している。——編集部，ASCII，245-260pp.

▶見えてきたカラー液晶
液晶表示をカラー化するうえで乗り越えてきた壁，および今後の課題について解説するレポート。——編集部，ASCII，8月号，309-316pp.

▶ビジネスマンの情報管理術
電子手帳用のプリンタ2機種の解説と，暑中見舞い状の打ち出しにチャレンジ。——塚田洋一，マイコン，8月号，313-316pp.

▶プログラムBASICカード
シャープが発売した電子手帳とパソコンとの接続システムを紹介し，さらにその手順と用途の可能性などについて述べている。——高橋雄一，マイコン，8月号，344-348pp.

▶なんでもQ&A MZシリーズ編
書院AX用のハード構成と，その辞書ROMとストリーマの仕様についての質問に答えている。——編集部，マイコン，8月号，391-392pp.

▶ビジネスショウ入門
晴海の展示会場の様子や，ちょっと目新しい出し物を紹介している。——編集部，LOGIN，13号，154-157pp.

▶図解世界のコンピュータちゃん第16回
ソニーのワークステーションNEWSと，データ記憶メディアについて解説。コンピュータ用語の独断的解説つき。——編集部，LOGIN，13号，174-175pp.

▶ネットワーカーホリック
電話代を安く抑えるためのパケット通信網について解説。PDSはX68000のボイスアラームを紹介。——編集部，LOGIN，13号，198-201pp.

▶GNUとストールマン
ソフトウェアの配布，利用についての新しい考え方，GNU(グニュ)プロジェクトの解説と，それを提言しているプログラマ，ストールマン氏を紹介。——編集部，LOGIN，14号，160-161pp.

▶図解世界のコンピュータちゃん第17回
SX-3を例にスーパーコンピュータの高速化などについて解説。コンピュータ用語独断的解説大事典もなかなか面白い。——編集部，LOGIN，14号，174-175pp.

▶ネットワーカーホリック
ゲームのサポートが好評のシエラオンラインBBS，リゾートネットの白浜シーサイドネット，八ヶ岳ネット，LACCOを紹介。PDSは，X68000のGROT.Xなど。——編集部，LOGIN，14号，198-201pp.

MZ-80K/C/1200/700/1500

MZ-700/1500

▶つきぬけた世界
幽体離脱を利用してオバケを退治しようというゲーム。——神前幸造，マイコンBASIC Magazine，8月号，126-127pp.

▶レーシング
敵の車や壁に当たらないように5周する。カーレースゲーム。——橋本隆，マイコンBASIC Magazine，8月号，128-129pp.

MZ-1500

▶Thunder'N'Blade
戦闘ヘリを操り，3ラウンドをクリアする。移植版。——深田郎，マイコンBASIC Magazine，8月号，130-131pp.

MZ-80B/2000/2500/2800

MZ-80B/2000/2500

▶影さんの山登り
坂の上から転がってくる岩を避けながら登って行くワンキーゲーム。——Calling，マイコンBASIC Magazine，8月号，132p.

MZ-2500

▶行軍将棋審判君
一瞬にして軍人将棋の勝敗を判定するショートプログラム。——柴田ガボン，8月号，I/O，168p.

▶STAR DUST
8種類の攻撃パターンを持つ敵と戦うシューティングゲーム。——電人，マイコンBASIC Magazine，8月号，133-135pp.

X1/X1turbo/Z

X1シリーズ

▶なんでもQ＆A X1/X1turbo/X68000シリーズ編
X1とポケコンのデータ転送，およびアナログジョイスティックの仕様，Sampling PRO-68Kの朗読機能についてなどの質問に答える。——編集部，マイコン，8月号，

参考文献

I/O 工学社
ASAHIパソコン 朝日新聞社
ASCII アスキー
インターフェイス CQ出版社
テクノポリス 徳間書店
ファミコン通信 アスキー
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIC Magazine 電波新聞社
LOGIN アスキー

“「天才の知恵」を万人のものにする情報科学”という大仰なサブタイトル。ここでいう「天才の知恵」とはコンピュータを設計する専門家のこと。万人というのはそれを実務で使うエンドユーザーのことです。つまり，ハードを設計する側とそれを使う側との断絶・ギャップが実務とかけ離れたアーキテクチャの計算機を生み，両者の仲立ちである SEやプログラマに大きな負担を強い，ひいては「SE30歳定年説」（優秀なSEをオーバーワークで潰してしまう）を生み出すことになるから，もっと使いやすいコンピュータが必要だ，と説いているのが本書なのです。著者はその解決をデータフローマシンに求めています。世の中はデータを主眼においた並列処理が中心だからです。しかし，前述のギャップが埋まらない限り，社会に必要なデータフローマシンは出てきません。著者はそれを何とかしようとして，本書を書いたようです。一種のエンドユーザーの啓蒙という感じですね。コンピュータを知る私たちが読むと，いささか甘口に思えてしまいますが，読者対象がコンピュータを避けている社会人ですからね。 （K）

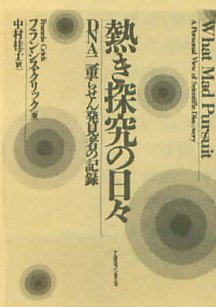
電脳的創造の方法 國井利泰著
日本実業出版社刊 ☎03(814)5161
A5判 232ページ 1,300円

389-390pp.
▶誌上公開質問状
熱転写プリンタCZ-8PC3，マウスCZ-8MN2について説明，紹介している。――多田太郎，マイコンBASIC Magazine，8月号，65-66pp.
▶ COSMO
敵を誘導して破壊するゲーム。自機に慣性がついている。――神谷栄治，マイコン BASIC Magazine，8月号，162p.
▶ FIRE FAIRY
妖精村の火の精が修業のために帽子で炎を受け止める。――細谷晴夫，マイコン BASIC Magazine，8月号，163-165pp.
▶ A-JAX ――ランキング――
ゲームミュージックプログラム，FM音源ボードとNEW FM音源ドライバが必要。――上田順一，マイコンBASIC Magazine，8月号，200-201pp.
X1turbo シリーズ
▶ SOFT RADAR
麻雀狂時代SPECIALパートⅡ，闇の壱与伝説，今夜も朝までPOWERFULまあじゃん2データ集などを紹介。――編集部，POPCOM，8月号，8-27pp.
▶ MONSTER ROAD
モンスターのいる道にお金がいっぱい。サイコロのない双六のようなゲーム。――中西弘幸，マイコン BASIC Magazine，8月号，166-168pp.
▶最新ゲーム徹底解剖!!
ソーサリアン「宇宙からの訪問者」を徹底解剖。――編集部，LOGIN，13号，122-125pp.

X68000

▶中古度チェッカ
SRAMの中に記録されている稼動時間の合計を読み出して表示させるもの。――磁石，I/O，8月号，134-135pp.
▶マウス感度変更プログラム
マウスの感度を好みに合わせて変えられる。これで狭い机でもOK(?)。――市原昌文，I/O，8月号，129-131pp.
▶ファイル/ディレクトリ削除ユーティリティ
指定したディレクトリを，その下にあるファイルごと一発で削除するコマンドを作る。――市原昌文，I/O，8月号，160-164pp.
▶ X-BASICの拡張プログラム3
拡張関数をXC用にライブラリ化する。――WIZARD N，I/O，8月号，183-192pp.
▶ MAKE & MKMK
作成日時など条件を指定すると該当するファイルのみをコンパイルするユーティリティと，その指示ファイル自動作成ユーティリティ。――井本裕司，I/O，8月号，217-234pp.
▶ AV WORKSHOP
Z'sSTAFF PRO -68 K Ver .2.0とZ'sTRIPHONY DIGITAL CRAFT，その他新製品の紹介。――宮本親一郎，ASCII，8月号，319-321pp.
▶ DSK
Human68k上でOS-9上のプログラムモジュールのディスアセンブルを行うセミオートディスアセンブラ。――川本琢二，ASCII，8月号，370-371pp.
▶ OS-9/68000徹底解剖（基本編）
OS-9/68000について，そのプログラム環境，ROM化の技法から最適化システムの設計まで，解析を試みている。――編集部，インターフェイス，8月号，140-229pp.
▶ GAMING WORLD
開発中のジェノサイド，38万キロの虚空などを紹介。――編集部，テクノポリス，8月号，23-40pp.
▶改造法苦楽部
X68000のアフターバーナーを，メッシュの煙からオリジナルのような煙に改造する。――石本淳，編集部，テクノポリス，8月号，105-106pp.
▶パソコンゲーム通信
プレイするたびに違った物語を楽しめるゲーム，ザ・キング・オブ・シカゴを紹介。――馬場総太郎（ゲーマーじいさん），ファミコン通信，14号，134-135pp.
▶ X68000ワールド
最新ゲームのジェノサイド，大江戸繁盛記，麻雀武蔵，WINGS，38万キロの虚空や，3DグラフィックツールのDIGITAL CRAFT，Musicstudio PRO-68Kのバージョンアップサービスのお知らせなど。――編集部，POPCOM，8月号，113-119pp.
▶ LET'S PROGRAMING
ガソリンの消費税込み給油計算のX-BASIC解答例。――藤本健，マイコン，8月号，270-279pp.
▶誌上公開質問状
X68000ACE-HDでフロッピーディスクから起動させる方法と，ボスコニアンのBGMの楽譜などについて答えている。――多田太郎，マイコン BASIC Magazine，8月号，66p.
▶移植版とりでの攻防
対人間，対コンピュータ戦が楽しめる。アイテムも強化されている。――軽澤菅，マイコン BASIC Magazine，8月号，169-171pp.
▶移植版 SOCCER
2人スポーツゲーム。ジョイスティックが2本必要。――LUM2，マイコン BASIC Magazine，8月号，172-174pp.
▶ THUNDER CROSS ――Great Battleship――
ゲームミュージックプログラム。――川野俊充，マイコン BASIC Magazine，8月号，187-191pp.
▶チャレンジ！ X68000
新着ゲーム，ファンタジーゾーン，ジェノサイドを紹介。――佐久間亮介，マイコン BASIC Magazine，8月号，276-277pp.
▶ NEW SOFT
異色のロボットアクションゲーム，ジェノサイドを紹介。――編集部，LOGIN，13号，16-17pp.
▶ X68000新聞
新着ゲーム，ニュージーランドストーリーや，プログラム言語FORTH，スプライトエディタTERAZZO，麻雀武蔵，サンダーブレード，ジェノサイドを紹介。――編集部，LOGIN，13号，176-181pp.
▶ SOFTWARE REVIEW
X68000版アフターバーナーを紹介，解説している。――船田戦闘機，LOGIN，14号，32-33pp.
▶ SOFTWARE REVIEW
デスブリンガーを紹介。――渡辺るね，LOGIN，14号，34-35pp.
▶最新ゲーム徹底解剖!!
新着ゲーム，ジェノサイドのステージ1，2を研究している。――編集部，LOGIN，14号，126-129pp.
▶ X68000新聞
ジェノサイド，ねじ式，デジタルクラフト，スターシップランデブー，大江戸繁盛記，C-TRACEトランスピュータ版，サイクロンExpress，MUSIC PRO-68Kのデータ集。――編集部，LOGIN，14号，176-181pp.

ポケコン

PC-1245
▶ FAST NUMBERS
16進数を使った反射神経ゲーム。――橋爪宏之，マイコン BASIC Magazine，8月号，177p.
PC-1600K
▶ポケットコンピュータ活用術
ポケコンの電子手帳化として，今回は電話帳機能の追加をしている。――塚田洋一，マイコン，8月号，317-321pp.
PC-E500
▶ DANGEROUS HIGHWAY
トラックの通る道の穴をふさげ！――水谷雅彦，I/O，8月号，264p.
PC-G801
▶誌上公開質問状
PC-G801のスクロール機能などについて解説している。――編集部，マイコン BASIC Magazine，8月号，65p.

熱き探求の日々
かのワトソン博士とともにDNAの2重らせん構造を発見し，ノーベル賞を受けた科学者フランシス・クリックの自伝。2重らせんの発見から遺伝暗号の謎が解明されるまでの，いうなれば分子生物学の古典時代におけるエピソードを中心に，筆者の体験などが語られる。「根っから科学好きのおじさん」が知的ゲームを楽しみながら遺伝の仕組みの謎に迫る様子には，科学や生物学について考えさせられる言葉がちりばめられている。
F.クリック著　中村桂子訳　TBSブリタニカ刊
A5判　272ページ　1800円　☎03(238)5931

人と機械の知能
本書では，知能とはなにかという問題について人工知能研究者による探求がなされている。知能は人間に固有のものか，動物にも認められるのか，機械で知能は実現できるのか。こうした基本的問題から，脳生理学の研究成果や人工知能分野としての認識，学習，言語について，さらにはヒトや動物の視覚とコンピュータの視覚処理まで，多岐にわたるテーマが展開されている。
M.A.フィッシュラー，O.ファーシャイン著
玉井哲雄訳　トッパン刊　B5判　400ページ
4640円　☎03(295)3461

書籍の価格は消費税込みです。

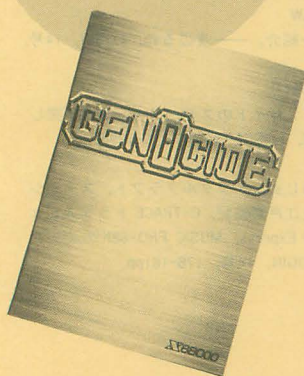
愛読者プレゼント

プレゼントの応募方法

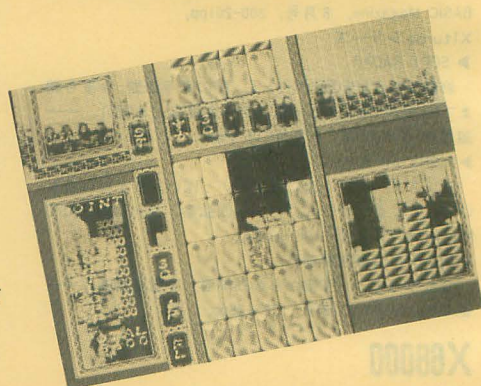
とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1989年 9 月18日の到着分までとします。当選の発表は1989年11月号で行います。

1 アンス・コンサルタンツ ☎092(522)6347
サイクロンExpress
X68000用5"2HD版　78,000円　1名

新しいアルゴリズムの採用でさらにスピードアップ，高度なテクニックとリアリズムを実現したサイクロンExpressを 1 名の読者に。

2 ズーム ☎011(613)0191
ジェノサイド
X68000用5"2HD版4枚組　8,800円　3名

バトルアクションゲーム，ジェノサイドを 3 名の読者に。次々と現れる敵をかたっぱしからなぎたおしていく痛快さをどうぞ。

3 アスキー ☎03(486)7111
琉球
X68000用5"2HD版　5,800円　3名

5 × 5 のフィールドにポーカーの役を作って点を競うパズルゲーム琉球を 3 名の読者に。沖縄情緒もたっぷり味わえます。

4 キングレコード ☎03(945)2111
ソーサリアン・スーパーアレンジバージョン
2,920円　3名

日本ファルコムのファンタジーRPGソーサリアンのBGMが13曲収められたCDを3名の読者に。

5 創美企画 ☎03(261)8631
コンピュータ・アニメーション Volume1,2
各4,800円　各2名

アメリカン・フィルム・インスティテュートのビデオアート部門でベスト・ビデオ・アワードを受賞したＬＤを各 2 名の読者に（Vol. 1 か 2 を明記のこと）。

7月号プレゼント当選者

1アドヴァンスト・ファンタジアン（大阪府）南光（大分県）矢野英明　2野球道（東京都）池田知繁　田中広幸（広島県）山中啓之　3ソングファイル68Kシリーズ　a.（北海道）小林敏哉　b.（神奈川県）田村明広　c.（千葉県）高橋政明　d.（栃木県）金沢光昭　43.5インチ2HDフロッピーディスク（愛知県）本多宗生　5図説コンピュータグラフィックス（北海道）田中宏征（石川県）千葉信幸

以上の方々が当選されました。おめでとうございます。品物は順次発送いたしますが，入荷状況などにより遅れることがあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

（価格はすべて消費税別です）

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

電子メモ2機種
PA-210/180
シャープ

シャープは，電子メモシリーズの新製品としてPA-210/180の2機種を発売した。価格はともに6,500円。両機種ともカタカナ12桁×2行表示で，メモしたデータを計算にも使えるデータメモ計算機能がついている。

PA-210は，スケジュール機能のほかに，名前8文字・番号12桁で記憶させた場合，最大450人分の電話番号がストアできる電話帳機能を持つ。名前のほか会社名を登録する項目があり，電話番号の呼び出しはどちらでもOKのダブルサーチ機能になっている。また，別売の通信用コネクタCE-210L(1,100円)を使ってPA-210同士のデータのやり取りも可能。

PA-180は，最大120件分登録できるスケジュール機能・電話帳機能のほかに豊富なアラーム機能を用意。スケジュールの日時や記念日などに合わせてアラームが鳴ると同時にそれらのデータを表示したり，毎日一定時刻に，あるいはスケジュールの10分前などに鳴らすこともできる。時計・カレンダー表示に週間スケジュール件数も表示される。

どちらもカードタイプでリチウム電池を使用。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221, 03(260)1161

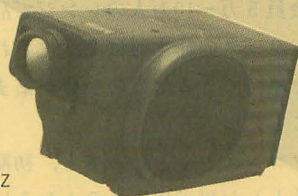
PA-180

（価格はすべて消費税別です）

最大100インチの鮮明映像が楽しめる
液晶ビジョンXV-100Z
シャープ

シャープは，液晶パネルを使った投影方式のプロジェクターXV-100Zを発売した。本体価格は485,000円。スクリーン・ビデオ・AVアンプなどと組み合わせてシステムを構成した場合のシステム価格は，ほぼ100万円前後。

この方式は，1個のランプから出る白色放射光をRGBの三原色に分解し，光シャッターとなる3枚の白黒液晶パネルを通過させ，さらにこの3つの単色映像をダイクロイック波長選択ミラーでひとつのフルカラー映像に合成して投影するもの。最大100インチから25インチまでの画面サイズを自由に選べ，スクリーンまでの投影距離を変えてもフォーカス合わせだけで鮮明な映像を得ることができる。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221, 03(260)1161

XV-100Z

プログラム関数電卓
fx-4500P
カシオ計算機

カシオ計算機は，学生やエンジニア向けのプログラム関数電卓fx-4500Pを発売した。価格は8,500円。

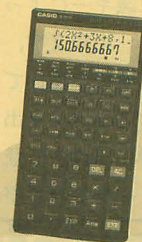
1,000ステップの容量を持ち，書き込んだプログラムをファイルネームで管理できるプログラム機能を搭載。分数・積分計算処理や124の関数/機能を持ち，三角・逆三角関数や双曲線・逆双曲線関数から座標変換にも対応しているほか，2/8/16進数の変換・計算機能，標準偏差・回帰分析の統計機能なども装備し，もちろん加減乗除やカッコの優先順位も判別できる。

また，電話番号やメモなどを入力しておけば電子メモ帳としても使用できる。

上段に数式，下段にその答えを表示できる大型2段表示。

縦形手帳サイズで幅73×奥行141.5×厚さ9.2mm，重量90g。

〈問い合わせ先〉
カシオ計算機㈱　☎03(347)4811

fx-4500P

超小型ワープロ
ワードバンクPen2
セイコーエプソン

セイコーエプソンは，従来機を機能アップした超小型ワープロ・ワードバンクPen2を発売した。価格は24,800円。

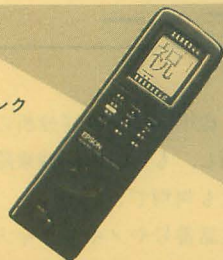
Pen2は入力したい文字を液晶画面上に手書き入力すればよいため入力のスピードが速い，外字登録を10文字にした，印字開始文字を指定すれば行の先頭からだけではなく途中からも印字できる，文字サイズを1行を一度に変更できる，などが新しい特長。JIS第1水準漢字はすべてカバーしているので変換の手間がはぶけ，またJIS区点コードによる入力も可能。さらに明朝体・ゴシック体の両方を標準で装備，印字部には48ドットの熱転写プリンタを採用し，カラーリボンも用意したのでカラフルで美しい印字が可能になった。

また斜体・網かけ・白抜き・白抜き斜体なども1文字ごとに指定できる。外字として手書きの文字・記号・イラストもそのまま登録できる。

サイズは横51×奥行32×高さ163mm，重量約200g。

〈問い合わせ先〉

セイコーエプソン㈱　☎0266(58)1705

ワードバンク
Pen2

手書き入力の電子手帳
EN-1
三洋電機

手書き入力専用の電子手帳EN-1が三洋電機から発売された。価格は22,800円。

漢字もかなも手書きでタッチパネルから入力でき，認識キーによってきれいな文字に変換される。記号文字，イラスト，JIS第2水準漢字も簡単な操作で入力が可能。

さまざまなデータを見出しのついたカードとして最大975枚まで登録でき，また項目ごとに分けられるので整理や検索も容易に行える。

また1週間分のスケジュールを時間単位で表示したり，スケジュールに合わせてアラームを鳴らすこともできる。

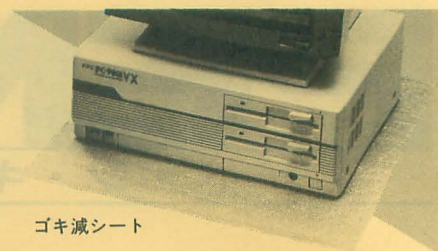
加減乗除・定数・べき乗・逆数・メモリ計算など多彩な10桁計算機能を持ち，さらにパスワードによるシークレット機能や表示の濃淡を変えたり入力音を鳴らしたりすることもできる。

〈問い合わせ先〉

三洋電機㈱　☎06(443)5144

EN-1

OA機器をゴキブリから守る
ゴキ減シート
エレコム

温度が一定で暖かく居心地のいいOA機器からゴキブリを遠ざけるゴキ減シートがエレコムから発売された。価格は1,500円。

ゴキ減シート

このシートを写真のようにパソコンなどの下に敷くと，アルミ蒸着などの光沢テープが乱反射するような構造になっているためゴキブリの感覚を混乱させてよせつけない。

サイズは縦50×横50cm，色はシルバー。

〈問い合わせ先〉

エレコム㈱　☎0723(62)7555

低価格A4ファクシミリ
ムラタM-5
村田機械

普及価格のファクシミリ，ムラタM-5が村田機械から発売された。価格は89,800円。

新製品はA4サイズの送受信が可能で，高速20秒伝送ができる。しかもボタン操作ふ

Again Watch

コンピュサーブがソースを買収
パソコンネットは日米とも2強体制へ

米国のパソコンネットの老舗であるザ・ソースをサービスする会社，ソース・テレコンピューティング社が，さきごろ同業界トップ企業のコンピュサーブ・インフォメーション・サービス会社に吸収合併された。これにより，米国のパソコンネットサービスはコンピュサーブとゼネラル・エレクトリック社のGEnieの2大ネット支配の時代に突入した。

ソース社は数年前から経営難が続いており，経営母体もクルクルと変わっていた。最後の経営者だったリーダーズ・ダイジェスト社も降りることになり，同業者であるコンピュサーブ社が吸収合併を提案，ソース側が了承したものだ。

これにより，ソースの会員はコンピュサーブ社に引き継がれることになり，現在，移行作業中。ソースは8月1日でサービスを打ち切り，ソースの会員は同日までに移行手続きを終了する。

米国のパソコンネットはトップがコンピュサーブで会員は50万人を超える。2位のGEnieには20万人の会員がおり，3位グループとしてデルファイやソースなど複数の企業が会員5万〜10万人のレンジで乱立している。ソースの経営難は高度化されたシステムを維持できるだけの会費が集まらなかったことにつきる。

パソコンネットといっても，初期のBBSを主体とした掲示板システムからは，かなり様子が変わっており，多彩な情報ネットワークになってきている。その反面で，会員個々から徴収する会費は少額なだけに，スケールメリットを追わざるを得ない性格のビジネスだ。ところがコンピュサーブと新進のGEnieの双方が他と比較にならないほど強い集客力を見せたため，それ以外のサービス会社は顧客獲得の目算が大幅に狂ってしまった。

この話は米国だけの話ではない。日本でも6万7千人の会員を抱える日本電気のPC-VANと会員数5万6千人に達したNIFのNIFTY-Serveが完全にアスキー以下の他社を大きくリードしている。ちなみにPC-VANはGEnieと，NIFTY はコンピュサーブと業務提携してるが，このことは決して偶然ではないだろう。

パソコンソフト流通会社が合併
日本SEが本格進出へ

合併といえば，独立系のソフトウェア・ジャパンと日本SE系のソフトウェア・インターナショナルの両パソコンソフト流通会社同士がこのほど対等合併した。ジャパン社は業界2位，インタ社は5位といずれも同分野では有力企業だった。

ソフト流通といえば，日本ソフトバンクが突出しており，後を4〜5社の中堅が追う形だったが，今回の2社合併により，バンクに対抗する勢力として浮上した。

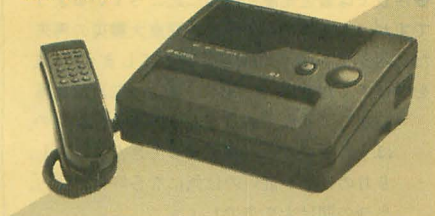
今回の合併劇は日本SEグループが本格的にソフト流通分野に進出，ソフトバンクへの対抗を名乗り出たと見るのが正しいだろう。これまでは比較的小さな市場だったソフト流通も，パソコンソフト市場の拡大にともない，早くも1千億円級マーケットに

たつと簡単な操作により誰でも操作できるよう配慮されている。また，専用電話機はオートダイヤル(ワンタッチ3，短縮17)採用の多機能型で，ワンタッチ再ダイヤルやオンフックダイヤルも可能。伝送速度は，9600/7200/4800/2400bps。

大きさは幅296×奥行244×高さ106mm，重量4.2kg。

〈問い合わせ先〉

村田機械㈱　☎075(672)8111

ムラタM-5

CG作品を集めたLD
コンピュータ・アニメーション Vol.1,2
創美企画

CGアニメの作品を収録したLD，コンピュータ・アニメーションVol.1,2が創美企画より発売された。価格は各4,800円。

コンピュータ・アニメーション

このアルバムは，1988年度アメリカン・フィルム・インスティテュートのビデオアート部門で「ベスト・ビデオ・アワード」を受賞している。リアルタイム・フライトシミュレーションを始めユニークなキャラクタによる3Dアニメーションがいっぱい。

〈問い合わせ先〉

㈱創美企画　☎03(261)8631

INFORMATION

'89全国(草の根)BBS大会
全国BBS大会事務局

全国のBBS局の運営社・会員が一堂に集い，交友を深め，情報を交換し，さらにさまざまな問題を討議する大会が，来たる8月20日(日)に東京・千代田区の科学技術館1F催し物会場で開催される。開催時間は午前10時から午後4時半まで。入場は無料となっている。

協賛企業・団体などの展示・実演・即売やBBS局へのフリーアクセスターミナル，不用品市場やゲーム大会などが計画されている。

〈問い合わせ先〉

全国BBS大会事務局　☎03(263)6940

新潟パソコン・ソフトウェアコンテスト
新潟発ハピネス
新潟テレトピア振興協会/新潟市ソフトウェア産業協議会

新潟市およびその近郊市町村からの応募者を対象に，ソフトウェアコンテストが来年2月18日(日)に開催される。

プログラム部門・アイデア部門とに分けて，パソコンソフトやパソコンを利用するアイデアを募集している。応募期間は，本年7月1日から10月20日までに応募票を提出，プログラム部門の作品は来年1月15日までに提出とされている。

〈問い合わせ先〉

ビット・ユニオン　☎025(222)1151

成長した。今後は有望な市場。とくにソフト開発と違って，純粋な商社活動であるだけに，この市場から得られる利益をめぐっての綱引きは今回の合併劇を機に活発化しそう。

ラップトップもカラー時代
NECが今月発売

先月号ではギリギリの入稿で東芝のダイナブックJ-3100SSを紹介できたが，続いて今度は日本電気からわが国初のカラー版ラップトップ機が発売された。

PC-9801シリーズのLX5Cという8色表示の16ビットの286マシンで，8月18日に発売。

価格は74万8千円。40Mバイトハードディスクを内蔵しているので，モノクロ版の5割増しと見なすのが妥当。

昨年秋にセイコーエプソンが開発発表をして以来，東芝，日本電気，日立製作所，三菱電機と相次いでカラー液晶パネルを用いたラップトップパソコンの試作品を公開してきた。その際，セイコーエプソンや東芝が提示した価格は百万円弱だった。モノクロ版の2倍以上が懸念されていたのだが，それから半年ちょっとたって，日本電気が最初に製品化したのだが，意外に安かったという感じだ。

これで価格的に法外な高さではない，ということが証明されたので，カラーラップトップパソコンが実用化に入る時期が早くも到来したと見ていい。16ビット普及品のカラー版が40万円前後で発売されればいよいよだ。東芝の20万円機も発売されたことだし，ラップトップパソコンも充実が著しくなってきた。テレビが見られるX68000のカラーラップトップ版にはお目にかかれるのだろうか？

スーパーファミコン，発売延期
ゲームボーイは好調のすべり出し

いつ出るか，との期待が高まっていたファミコンの16ビット後継機スーパーファミコンが今後1年間，発売凍結になった。理由は輸出用ファミコンとゲームボーイの需要が好調で，メモリチップ(主としてソフト用のマスクROM)が不足。スーパーファミコン用までないことによる，と任天堂では説明している。

スーパーファミコンの仕様は16ビットのカスタムCPUで，メインメモリ管理能力14Mバイト，カラー256色表示。

ちなみにゲームボーイは7月までに60万台を出荷。年内140万台を国内で，8月からスタートした対米輸出では年内150万台を予定。来年秋には内外合計で累計1千万台の大台乗せも可能だという。

ディスカウント続くFD
3.5インチ2HDも300円台突入へ

使っている人には当たり前の話だが，これまでフロッピーの中で唯一，高値販売が続いていた3.5インチの2HDタイプ品がいよいよ400円を割り込んだ。

5インチ品がタイプを問わず暴落しているのはもちろん，3.5インチの2DD用がワープロ用として売れており1枚200円台まで安くなっている。いよいよ2HDも安くなり，FDの安さが徹底してきた。朗報ではある。

(K.T.)

FROM READERS TO THE EDITOR

今年の夏はムシ暑いわ，台風がボコボコ来るわで（これって，関東地方だけなのかな），ずいぶんと気合の入った“夏”になってしまいました。でも，Oh！X読者の皆さんには夏バテなんて関係ないよね。今月も元気いっぱいにいきましょう。

◆グラフィックを特集してほしかったので，とても7月号の特集は楽しく読むことができた。コンピュータ内部構造を知り尽くしたユーザーは少ないので，もっと高級な利用法や，機能法を示すプログラムの例をたくさん載せてほしい。
岸野　真樹（18）香川県

◆コンピュータグラフィックにはとても興味があったので，7月号の特集はためにもなりました。とても面白かったです。
浜田　憲一（20）埼玉県

◆特集の「3Dグラフィックへの飛翔」はとてもよかったと思います。多少，頭痛がするのを除けば。　清水　達朗（20）岐阜県

いきなりZバッファアルゴリズムですから，ちょっと難しく感じられるところがあったかもしれません。でも，理論と処理方法の概略だけでも理解していただければ，後々きっと役に立つことができる特集だと思っています（少し自慢）。

◆7月号は難しい！　DōGA・CGAシステムほしい！　合体，変形，ファンネルひゅんひゅんてのが作りたい!!　土井　宏治（15）東京都

今月のDōGAの連載では，変形するところまではいったみたいですよ，ファンネルは付いてなかったみたいだけど。

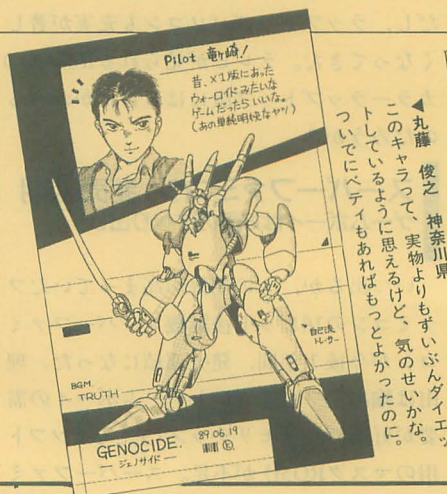
◆「X68000マシン語プログラミング入門」へリクエストなんですけど，X68000のウリはやっぱりグラフィックかスプライトだと思います。だから早くグラフィックのことを取り上げてください。村田敏幸さん頼みます。
牛島　睦（17）福岡県

◆「X68000マシン語入門」に続いて，今度は「X-BASIC」の連載が始まったが，これはなかなかいいことだと思う。長期連載で基礎をみっちりとたたき込んでほしい。グラフィックやFM音源なんかよりも，こういった標準的な入出力のプログラムのノウハウを連載してほしい。毎月こういったかたちで見ているほうが，一度にマニュアルを読むよりよく頭に入ると思う。
斎藤　健一（22）香川県

ひとつ説明すると，すぐに小さいプログラムで実践してみる。この積み重ねがコンピュータを使って勉強するのにやはり最適なのかもしれませんね。村田，泉両氏にもみんなの期待に沿うよう，“締め切り守って”がんばってもらいたいものです（ついつい本音が……）。

◆「マシン語カクテル in Z80's Bar」を読んで，創刊時のパッケージソフトの紹介を思い出し（なんだか構成が似てるように感じた），久しぶりに創刊号を読んでみました。古きよき時代ですね。ついこの前のようにも感じますが，時の流れは早いもの。Z80のニーモニックを覚えようとして，何回も挫折したことを思い出しながら楽しく読ませていただきました。
栗坂　明（20）埼玉県

◆ややっ！　MZペンギンが復活しましたね。ということは，そのうちマジックバスのアニメ講座やアイドルインタビュー，SBS，それに必殺のオークスターのお姉さんの表紙も帰ってくるんですね！　山川　俊介（21）徳島県

オークスターのお姉さんは帰ってこない代わりに，8月号には“牛”を登場させときましたけど，いかがだったでしょうか。

◆最近，シャープの電子システム手帳を買ったんですけど，これはなかなか使えます。特にメモ機能がいいのです。分類コード方式になっていますから，必要な情報が素早く取り出せます。また，ICカードも最近はゲームなんかも出始めていいと思います。中山　美喜也（28）岩手県

あのTETRISもビー・ピー・エスから発売されるようだし，もうすぐ発売となるStationary PRO-68Kを使えば，X68000にもつながるみたいだし（それもジョイスティック端子に），これからOh!Xでも電子システム手帳は注目したいと思っています。

◆伊豆では最近地震が頻繁に起こっているようですが，大丈夫でしょうか。関東大震災の再来だと困ります（Oh!Xの発売が遅れてしまう）。気をつけてください。　菅本　淳二（22）広島県

心配してくれてありがとう。でも，伊豆のほうもずいぶんとおさまってきたようだし，9月の10日が近いのは気になるけど，まっ，当分の間は大丈夫でしょう。

◆シャープさんにお願いがあります。独自の64ビットCPUを出してください。シャープさんには作れるはずです。待ってるけんね。名づけて，MPU-SX64（光ファイバー方式）。球形のCPUだよ～。　相賀　清（21）岡山県

◆いま，X1turboZⅡを買うか，冬のボーナスと合わせてX68000を買うか迷っています。う～ん，turboのソフトを捨てるのはもったいないし，X68000も捨てがたい。宮地　英則（25）兵庫県

◆つ，ついに，響子さん24枚組が届いた!!　これでしばらくX68000はお休みになってしまうでしょう。LDPを酷使するなぁ……。話は変わりますが，編集部の方で「ほーほほほ！」と高飛車な笑い声が聞こえてくる，あの「白鳥麗子でございます」のファンの方はいませんか？　いまやあのマンガは，時代の“大”トレンディであります。　山崎　勝義（35）茨城県

白鳥麗子って，スピリッツのツルモクに出てくる彼女だったら知ってるけど，こちらの白鳥麗子さんは（U）氏のほうが専門なもので，今後読んで勉強しておきます。

◆中日，奇跡の逆転優勝!!
金城　起賢（17）神奈川県

◀丸藤　俊之　神奈川県
このキャラって，実物よりもずいぶんダイエットしているように思えるけど，気のせいかな。ついでにベティもあればもっとよかったのに。

◀鶴見　明子　東京都
このイラストの意味がわからない人は，先月号を見てね。こうなりゃ「私は女だシリーズ」で連載にでもしますか。ねえ，鶴見さん。

愛知県人会の荻窪圭氏は，「中日ファンはいさぎよく今年は玉砕するんだ」と騒いでおります。

◆免許はあるが，車がない。電脳倶楽部はあるが，X68000がない。こんな僕に愛の手を。

陣内　達夫（19）大阪府

ついでに，仕事は山のようにあるけど，時間がない。という，某編集部にも愛の手を誰か差し伸べてくれないかな……。

◆7月号163ページの渡辺仁さんへ。私も3歳と2歳の息子に付き合って，「ふえはうたう」，「ピコピコポン」をはじめ，「やっぱりヤンチャー」，「プルプルプルン」，「のんたっく」，「あいうえお」など，教育テレビの幼児番組をよく観賞しております。ちなみに私は「おかあさんといっしょ」の神崎ゆう子お姉さんの隠れファンです。　平沢　昭介（30）千葉県

結局は若いお父さんたちも，結構楽しみながら，幼児番組を見ていたりするわけですね。

◆げげっ，斎場スティック君って，まるで俺のことみたいじゃん。大学受験のときは情報工学科しか受けなかったし，将来はタイトーに入社しようと心に決めてるし，実は自宅から自転車で5分くらいのところに知る人ぞ知るタイトーの中央研究所があるのだ。スティック君ほど工作技術はないけど，ZUNTATAの方々，3年後の就職活動のときはまっ先に行きますのでそのときはよろしくお願いします。

髙山　稔（20）神奈川県

俺のことみたいっていったって，まさか小学校の自由工作で火縄銃なんか作ったりはしていないでしょうね。

◆スプライトエディタの「TERAZZO」を間違えて，「テラゾー」などと読んでいたのはこの私です。今月の中森氏のレビューを読むまで1カ月もマヌケな読み方をしていたかと思うと，情けなくなる今日このごろ。

河部　義信（17）愛知県

大丈夫です。編集部でも最初は，みーんな「テラゾー」と呼んでたみたいですから。

◆暑い。とにかく暑い。今日なんて31度ぐらいもあった。ずっと前のOh!Xに載っていた簡易FM音源/PSGミキサーもハンダが溶けて動かない（これはウソ）。8月になったらいったいどうなってるんだろう？　上地　雅之（17）沖縄県

沖縄も暑いけど（去年の夏は万座で遊んでました），東京の朝から晩までムシムシした暑さに比べると，まだ許せるような気がします。

◆誰かになにかを説明しようとするときに，言葉より絵を使ったほうがわかりやすいということは，ことわざにも似たような言葉があるように明らかだが，3次元モデルをわかるように絵にするのは結構たいへんなことだ。やはり究極の伝達手段は，直接脳に電気信号に変えて送り込む方法なのでしょうか。

田中　雅俊（22）神奈川県

別に電気信号送り込むのもいいけど，Oh!X

▶藪田　俊平（18）和歌山県
リアルな戦闘機のイラストなんて，最近珍しいよね。この調子で，お次はジェットヘリでもリクエストしちゃおうかな。

▶味野　真一　群馬県
起動するとこんな子が微笑んでくれると暑さもふっ飛びそう。でも，風疹と熱暴走の組み合せは，ちょっと危ないような気が……。

1冊分くらい入れちゃうと，すぐにオーバーフロー起こしちゃって，頭から煙モクモクなんてことになりかねませんよ，きっと。

◆また今年も花粉症に悩まされている。昨年はたいしたことなかったが，今年は半端ではなかった。これで6年連続だぜ。編集部にも花粉症の方いませんか？　なにかいい防止法があったら教えてください。　吉田　裕（17）岩手県

これといった防止策はないようですが，某「ストナリニ」という鼻炎用の薬が，社内の花粉症仲間（？）の間では人気があるみたいです。

◆サイバースティックを買おうと思う。ついでにフットペダルもほしいし，ハンドルもほしい。コーヒーのサービスなんてのもいいと思う。

上野　民夫（21）東京都

ハンドルはぜひともほしいところですね。それとA-JAXやサンダーブレードの発売に合わせて，原寸大のヒューイコブラをセット販売する。ついでにお持ち帰り特価販売もやる。やはりこれで決まりでしょ。

◆最近UNO（総理大臣ではない）というカードゲームを初めました。やってみると結構楽しめるので時間を忘れます。ところで，このUNOのX1版のゲームソフトをどなたか考えていただけないでしょうか。以前，MZ-2500版は発表されたと思うので，移植の好きな方お願いできないでしょうか（もちろん，編集部の皆さんでも結構なのですが……）。　加藤　徹（27）愛知県

◆ディスプレイにフィルタを付けようと思うのですが，「紫外線をカット」とか「電磁波をカット」とかいわれてもよくわかりません。紫外線や電磁波などは，目にどのような悪い影響を及ぼすのでしょうか。水谷　邦久（17）神奈川県

そういえば，サングラスまで昔は発売されていたようだけど，最近はあまり騒がれなくなってしまいましたね。でも，やはりそれより先に気をつけなければならないのは，視力の低下でしょう，きっと。

◆早くも夏休みをとって家族3人で軽井沢に旅行してきました。初夏の軽井沢はまだ人通りが少なく別荘の一帯の森では，カッコウやコジュケイや，なんとウグイスの鳴き声も……。しばし，都会の埃を落とし，とてもさわやかな気分に浸ることができました。

国見　協一（32）神奈川県

◆もうすぐ前期試験が始まる。1年生のときは，1週間で14教科（1日2教科）もあったが，2年生のいまは，1週間ちょい（8日）で4教科しかない。うーん，これは困った！　4教科しかないのはいいのだが，あとの教科は後期のみの一発試験だ！　200ページぐらいある本1冊分が試験範囲なんて，ちょっと"酷"だと思いませんか？　さー，勉強するか!!

西村　泰和（21）千葉県

◆このたび，めでたく普免が取れました。それはいーんですが，僕は背が低いうえに童顔なので，道ですれ違うほかのドライバーの目が妙に気になってしまいます。ドライバーの皆さん，僕はちゃんと免許持っているんですから，変な目で見たり，にらんだりしないでください。

望月　隆史（18）長野県

◆「えっ，我が家の水道代が35万円？」。もちろんミスであった。そのあとお偉さん2人が詫びを入れに来たが，このとき持ってきた手土産が「千里の水」なる缶飲料（？）。文字どおり千里の水が入ってるやつで，水道局で作っているものらしい（もちろん非売品）。飲んでみるとほんとにただの水で，缶に書いてある原材料の欄を見ると「水道水」。どーりでカルキ臭いわけだ。缶には太陽の塔なんかの絵（万博公園だな）が印刷してあって，「ゆめ・わくわく」とか書いてあった。どーゆー意味やねん。

四ッ谷　類（15）大阪府

水道料金が35万円っていう話も凄いけど，缶入り水道水って，いったいなんなんでしょうね。蛇口をひねればどこでも飲めるものなのに。

◆6月中旬の夕方，秋葉原駅のホームでパソコンサンデーに登場していた髙橋雄一氏らしき人を見た。街中で有名人に会うのは，2年くらい前，電車のなかで沢田聖子を見て以来だったのでウキウキしていたのだが，本人と確認する間もなく彼はどこかへいってしまったのであった。ウーム，残念。　望月　隆（24）東京都

◆6月18日（くもり）。今日は東京おもちゃショ

ーに行きました。会場に着くと，人人人で，大の大人もたくさんいました。やはり自分としては，NECとSEGAのブースが気になり，いちもくさんに行ってみると，なんと，メガドライブのアフターバーナーが出るそうで，今日はそれだけで満足して帰りましたとさ。

平野　岳志（18）東京都

◆メガドライブを買っちゃったい。けど，ソフトが少ないので，メガアダプタまで買っちゃった。同じ68000CPUなんだから，早く同じソフトが出揃えば，Oh!Xを読んでもくやしい思いをしなくなるぞ。　高橋　克公（33）岩手県

平野君のお便りにもあるように，今度はアフターバーナーも登場するそうです。メガドラ版のできに期待したいところですね。

◆先日，スタークルーザーのX1turbo版を解きました。あのストーリーのクサさはイースよりスゴイ！　しかし，よくあんなのがturboで動くものだと感心した(少し遅いけど……)。それから話は変わるが“画餅”をX1にも発表してください。　岡野　礎（17）静岡県

◆イースⅡのエンディングで，ゴーバン（？）がいっていた魔法についての言葉を読みながら，このなかの魔法はいまの科学や文明というものじゃないかと思った。人は，暮らしをよくするために機械を作り出し，その公害で滅びかかっている(らしい)。友人に話したら，考えすぎだといわれたけどね。　川合　弘之（18）東京都

イースの場合だと，結構，そういった意味を隠し持っていたとしてもおかしくはありませんよね。

◆現在，僕は友人の家に「戦国群雄伝」をやるために毎週通ってます。僕にとってシミュレーションでこんなに面白いゲームは初めてで，感動しまくる毎日です。信長や家康が強すぎると思うのですが，ひいきの毛利も西日本一の強さを誇っているので，あまり気になりません。光栄さん，今度は「三国志」でこーゆーのを出してください。待ってますヨ。

島村　庸也（16）広島県

◆神戸製鋼のCMはいい。お次はスナフキン復活か!?　木村　陽一（18）京都府

◆3Dグラフィックですか。X1turboのグラフィックもせめてMZ-2500ぐらいほしいですねぇ。やっぱりX68000はいいですねぇ。K氏の絵（アニメーション）はよかった。僕はあのF1の傾いている感じがとても気に入ってしまった。早くX68000が買いたい。う～ん，あと20万円ちょっと足りないなぁ(現在貯金4万円ぐらい)。つらいものがあります。なんでOh!Xのスタッフの人はみんな持っているんだよー。持っていない人いますか？　西川さんも高沢さんも荻窪さんも祝さんも(『試験に出るX68000』なんて出さないでくださいよ！)，みーんな持っているだろー。くやしーっ。と，なげく貧ボー人でした。

樋口　雅人（16）福島県

持っていない人も結構いますよ。パソコン貧ボーはもっとたくさんいるみたいだけど。

◆最近，「部屋がコードのジャングルと化している」というハガキを見受けるが，まだまだ甘い！私なんぞは下宿が6畳だっつーのに，電源用ケーブルだけで20本！（当然，冷蔵庫，時計などの生活家電は除いている)。これらはすべてX1 turbo，AV，シンセなどの命綱である。各種インタフェイスやラインケーブルなどはいったい何本あるんだろう。なかには借りものも入ってるハズなんだけどな。TEACのAVP25がひとつで足りなくて，もうひとつ買い足すくらいだもん。よくヒューズが飛ばないなぁ？　ちなみに，AVP25とは15口のディストリビュータです。

佐藤　仁（20）山梨県

ほんと，よくヒューズが飛ばないものですね。でも，こんなになっているんじゃ，ケーブルが群生している姿って，かなり壮絶なものがありそう。

◆うちのマイコン部は，部員約50名のうちシャープ派が3人しかいない。よってシャープのマシンはいままでなかった。しかし，やっとのことでX68000が導入されることになった。次はOh!Xの導入だ。　長井　勝栄（20）高知県

あの一，できれば毎月部員の皆さん全員に買っていただければ，とってもありがたいんですけど……。

◆1987年2月号のOh!MZに，「データの互換性を探る」と題してBASICを媒介に各種ワープロの文書ファイルを相互にコンバートする記事が載っていましたが，スーパー春望Ⅱや将軍など，新しいワープロ用のプログラムを開発していただく予定はないでしょうか？

田代　勝久（19）佐賀県

田代君は，いったいどのマシンと文書ファイルのやりとりをしたいのかな。この2つのワープロは，もしX68000が相手の場合だと，2HD版のMS-DOSのテキスファイル形式にすることができれば，少し面倒だけどなんとかコンバートは可能だと思います。

◆ああ気持ちが悪い，気味が悪い。いったいなんのことかといいますと，実は僕はOh!X1月号のプレゼントに当たり，その本がつい先日届いたんです。ここまでなら「ああ，嬉しいなぁ」なのですが，それから約1週間後，なんと今度はR-TYPEが届いているではないですか！　封筒をよく見ると，某LOGiNの文字！　そうです，こっちでもプレゼントに当たってしまったのです。しかし，2度あることは3度ある。その次の日には僕の大好きな永井真理子お姉さまの直筆サイン入りテレカが速達で届いたのです。もう今度は驚きませんでした。背筋を走る冷たいものを感じただけでした。私は花の受験生（現役だよん）なのです。こんなことで運を使ってしまっていいのでしょうか。ウーン複雑な気持ち。　橋本　浩二（18）兵庫県

◆6月24日，「スーパーフェスティバル in DOME」でウインクのミニコンサート。7月8日は武道館で浜田麻理。そんでもって10月27日からは幕張で東京モーターショウ。あー，コンサートは最高だったし，モーターショウが待ちどーしい。こんな私は浪人生……，さあ夏が勝負だ！　大河原　知道（18）神奈川県

橋本君や大河原君みたいな元気な人たちが，受験雑誌としてのOh!Xを支えてくれる原動力なんでしょうね，きっと。

◆いま，僕の姉がシャープの入社試験を受けています。もし，シャープの社員になれたらパソコンが安く手に入るかなあ。安く手に入るようだったらX68000 PROがほしいよー。

三上　昌浩（16）島根県

◆田山真美子ちゃんのデビュー曲「青春のEVERGREEN」買ってね。8月21日発売だよ。

滝田　邦博（19）千葉県

◆Oh!Xの読者の方や編集部の皆さんのなかで，岡村孝子さんのファンの方はいらっしゃいますか。先日，「Eau Du Ciel」というアルバムが出ましたので，よかったらぜひ聴いてみてください。そのなかでも2曲目の「虹を追いかけて」と7曲目の「愛がほしい」はお勧めです。友人に孝子さんの歌は皆同じに聞こえるという人がいますが，私はよく聞けばそんなことはないと思います。ささやくような歌声はいつ聞いても気持ちをフレッシュにしてくれます。

根本　忠之（21）千葉県

◆今日は念願のX68000が我が家に来る日です。がんばれ，来月から1年ローンだ!!

横山　昭男（20）高知県

それはオメデトウ。ついでにローンもがん

▶堀　幸司（19）福岡県
サイオブレードはなかなかよくできているから，メガドラに移植されると，きっと楽しめる作品になると思うんだけどね。

▶岩本　智雄（17）埼玉県
なかなかに夏らしいイラストだけど，これって「猫に小判」じゃなくって，「ＴＡＭＡに金魚」っていうのかな。

ばってね（なんのこっちゃ)。

◆私は初心者です。根っからのビギナーです。私にもわかりやすいX68000を生かす方法をこれからもよろしくお願いします。

堀　耕一郎（27）大阪府

◆いま仕事場にX68000を置いているのですが，自宅にもX68000を置かざるを得なくなってしまい，困っています。家庭に仕事を持ち込むのはイヤなのですが，いつでもX68000が傍らにいるという環境はまんざらでもないけど。

伊藤　秀樹（30）愛知県

◆X68000関連の記事が読みたいばかりに，ログイン，コンプ，ポプコムを買っているが，どれもX68000コーナーが少ない。Oh!Xはシャープのパソコンの専門誌なのに，特にゲーム関連の記事がもの足りないので，ついつい他誌も買い込んでしまう。どうか，Oh!Xだけ買えば，他誌は不必要！　くらいいえるような雑誌になってほしいものです。プログラムの記事は，いまのままでもダントツなんだから。

田中　忠昭（26）東京都

「Oh!Xさえあれば他誌は不要」と呼ばれるように，ぜひなってみたいものですが，そのためにはあと500ページくらい厚くして，スタッフの数もあと100人くらい増員しなきゃ，きっと無理な話でしょうね。

▶丸山　晃則　静岡県
確か丸山君は兵庫県に住んでたはずだけど，いまは富士の裾野でメカトロニクスの研究をやっているわけね。これは将来が楽しみ。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目（売る・買う・氏名・年齢・連絡方法……）を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集室では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲　間

★MZ-700/1500，X1その他の機種を対象とした「EXTRA」では会員を募集します。2カ月に1度発行の会報には，お便りや短いプログラム，Q&Aに売り買いのコーナーなど豊富な情報を満載（A5判サイズで4～16ページ)。入会金は100円。会報1回分の会費は112～143円です。現在会員数は49名で，売り買いのコーナーは会員以外の方にも開放しています。興味のある方は100円分の小為替を送っていただければ案内書と会報をお送りします。また，案内書のみご希望の方は，62円切手同封のうえ封書にて連絡を。　〒811-42　福岡県遠賀郡岡垣町戸切794-3　筑紫高宏（22）

★「Game Land」というサークルを発足させるにあたり会員を募集します。機種は一切問いません。活動は，ゲーム情報を中心にさまざまなことを行っていきたいと思っています。詳しいことは62円切手同封のうえ封書にて。　〒929-11　石川県河北郡七塚町浜北ホ26　西田豊（16）

★MZ-2500ユーザーを対象としたサークル「MCJN」では会員を募集します。活動内容は月1回発行のディスク会報を中心にさまざまなことをやっています。現在会報もVol.8まで出しています。詳しいことは62円切手を同封のうえ封書にて連絡を。　〒921　石川県石川郡野々市町粟田1-147フレンドリーハウスヤマグチ207　相田寛（21）

★パソコンサークル「IPL」では，Xシリーズのパソコンユーザー会員を募集します。活動内容は，会誌「IPL.X」の発行を中心に，ゲームも含めたコンピュータの活用法などについて考えていきたいと思っています。詳しいことは72円分の切手を同封のうえ連絡を。直ちに「IPL.X」の創刊準備号を送付します。また，同時にイラストの描けるスタッフも募集していますので，イラストに自信のある方は大歓迎です。　〒300-12　茨城県牛久市牛久町1137-36　柳舘孝太郎（18）

★X1turboユーザーを対象とした「Studio Attic」では，その筋の方やその筋を目指している方々の会員を募集します。活動は，ディスク会報の発行を中心に行っています。興味のある方は，120円分の切手を同封のうえ封書にて連絡を。折り返し会報のサンプル版をお送りします。　〒581　大阪府八尾市末広町2-1-2　磯崎元洋（16）

★X68000ユーザーの方で，情報交換やPDSの交換などを行ってくれる方を探しています。サークルではないので気楽にお付き合いいただける方を希望します。賛同される方は62円切手を同封のうえ，連絡を。　〒533　大阪府大阪市東淀川区大桐2-1-19　笠原聖一（19）

★X68000ユーザーを対象としたサークル「達人」を発足させるにあたり，会員を大募集します。活動内容は，月1回のディスク会報と会誌の発行を中心に，ゆくゆくはD-RETURNを超えるようなソフトの製作を考えています。「プログラムなら任せろ！」という人や，「アイデアならバッチリだぜ！」という人など，とにかく「やってやるぜ！」と思っている人は，62円切手同封のうえ封書にて連絡を。　〒433　静岡県浜松市初生町1043-4　阿部猛美（17）

★「Thank」では，X68000ユーザーでプログラムを自分で組みたいと思っている方，ゲームの好きな方，音楽の好きな方，CGを楽しみたい方を募集します。活動はプログラムの解説，ゲーム特集，ミュージック講座，CGコーナーなどを盛り込んだ月1回発行のディスク会報を中心に行っています。入会ご希望の方は62円切手を同封のうえ封書にて連絡を。　〒377-02　群馬県北群馬郡子持村田尻1789-2　新藤満春（18）

売ります

★MZ-2521用ビデオRAMボード・パレットボード・ボイスボードの3点セットを1万5千～1万8千円で。連絡は往復ハガキで。　〒737　広島県呉市的場2-3-11　上瀬彰（43）

★ディスプレイMZ-1D26（アナログ/デジタル，400ライン）の箱付きを4万円で。また，ポケコンPC-1360K（箱，マニュアル付き）を2万円で。連絡は電話番号明記のうえ往復ハガキで。　〒733　広島県広島市西区高須2-2-50　藤山隆之（20）

★プリンタCZ-8PC2（ケーブル，マニュアル，リボン，箱付き）を3万円で。連絡は往復ハガキで。　〒505-01　岐阜県可児郡御嵩町伏見593-1伏見ハイツ203　山田勝人（24）

★MZ-2500用RAM DISK・RM-25E（テレシステムズ製，640Kバイト，付属ソフト付き）を送料込み2万5千円で。連絡は往復ハガキで。　〒969-16　福島県伊達郡桑折町字西大隅76　半沢勉（19）

買います

★X1用ドットインパクトプリンタCZ-8PDシリーズのいずれかを送料込み1万円前後で（ケーブル，マニュアル付き，箱なし可)。連絡は往復ハガキで。できれば手渡し希望。　〒259-12　神奈川県平塚市長持273-2　大森幹雄（17）

★X1用カラーイメージボードCZ-8BV1を送料別1万円前後で。連絡は価格明記のうえ往復ハガキで。．〒992　山形県米沢市福田町2-3-58　白銀浩和（18）

★カラーイメージユニットCZ-6VT1（付属品込み）を3万～3万5千円で。また，CZ-600C用増設1MB RAM・CZ-6BE1を1万円で。連絡は往復ハガキで。　〒771-14　徳島県板野郡吉野町西条字地免138　大浦晃裕（20）

★FDD・CZ-520F（付属品込み）を5万円前後で。連絡は希望価格明記のうえ往復ハガキで。　〒940-21　新潟県長岡市上富岡町21おふくろ荘101　太和田誠（18）

★X1turbo model 10用G-RAM・CZ-BGR2を送料込み1,500円で。完動品であれば説明書がなくても可。連絡は往復ハガキで。　〒010　秋田県秋田市濁川字三枡作1-74　岩沢正樹（16）

★CZ-600DE用チルトスタンドCZ-6ST1Eを3千円前後で。連絡は往復ハガキを希望。　〒620　京都府府福知山市かしの木台5-54-4　伊井英善（16）

編集室から from E・D・I・T・O・R

DRIVE ON

▼「グラフィック環境の課題」はわかりやすく，かつポイントを押さえています。CGがそこここに氾濫してはいるものの，いざ「やってみよう」と思ってもそう簡単にはいきませんしね。3Dとなればなおのことでしょう。しかし逆に「やろう」と思えば「できる」環境が今のわれわれには用意されているのも確かです（特にX68000ユーザーさん，君のことだよ)。8ビット全体のパソコン市場では，高度なCGを作ることにおいては「翼がなかったのではなく空がなかった」というのにはうなずきました。それから，3Dデータの共通化などはぜひやってもらいたいことです。2Dでもやってほしいなあ。それにしても，これからは16ビット以上でないと3DCGの話にはついていけないようですなあ。ワイヤーフレームでデモってたシルフィードやウィバーンはいまさらながらに凄いと思います。スタークルーザーは8ビットの現役ぶりを物語っているし。まだまだ発展する分野ですね。
大津和之（19）X1turboZ 福岡県

▼「グラフィック環境の課題」は，短い記事のためかどこか消化不良的で，もっと述べたいことがたくさんあったのに出さずじまいだったような印象を受けた。せっかく貴重な問題提起をしているのだから，できればここでも具体的な事例を多用してCG最前線などまでやってほしかった。「3次元データ処理の基本技」での最も重要なポイントは多面体の回転・移動だが，このあたりは簡潔に流してあって多少物足りなさを感じる。公式集的な記述のみではたとえリストつきでも「基本」を説明したことにならないのでは。そうした説明不足を「透視変換アルゴリズム」までできて補っている感じ。「Zバッファアルゴリズム」では図などがよく工夫されており，これらの図と囲み記事とでも考え方はよく理解できそうだ。ただ，全体的にかなり駆け足なのでもう少し書き込んでほしかった。話は変わるが，研究用モードつきの強力な「森田の将棋II」がX68000用に出た。このソフトのためにX68000を買いたくなってしまった。X1用のほうは処理が遅く，定石で攻めまくると弱い部分も見られたが，このへんはX68000のパワーでなんなく解決しているだろう。プロでも勝てるかどうかという難しい局面が用意されていることを期待する。
中野賢一（29）MZ-2000，X1G/turboZII，PC-1251，PC-8801，FM-8，B16LX 山口県

▼「透視変換アルゴリズム」は全般に平易に説明されていると思うが，座標系の回転のところはもっと詳しい説明があったほうがいいのではないだろうか？ X軸まわりの回転も考慮し，各軸まわりの回転の順序を変えるとどうなるか，などなど。「Zバッファアルゴリズム」も十分に解説されている。8月号の続編にも期待する。特にリスト3の汎用サブルーチンはコンパイラの種類を問わず使えそうなので，私にとってはたいへん有益なプログラムだ。また「X-BASICプログラミング調理実習」は，全般に平易すぎると思う。「X-BASICの起動」などは不要。第1回の内容としてはこんなものかもしれないが，今後はグラフィックと音楽関係をやさしく書いてくれるといいと思う（HuBASICとの比較なんかもちょっと載せてもらえると助かる人も多いのでは)。
森川一（24）X1turboII，X68000ACE-HD 北海道

▼「スムースシェイディングへの道」のように「最低限の努力で最大限の効果」的な手法は非常に好感を覚える。「どんなに細かくデータをとって3D表示したって人間の目って結構鋭いんだよね。でもこうやって人の目にやさしくすると案外粗くても気がつかないんだ」というのが友人の弁。なるほど，人間の人間たるところですね。
小笠原陽介（21）PC-9801EX2，PC-E500 東京都

▼画餅システムのような発想の（つまり成長する）グラフィックエディタはいいですね。これまで発表されたものは，それ1本で完成を目指しているためオプションプログラムの設定まで気を回してはいませんでした。その点，画餅は欲しい機能だけでも動作するし，バージョンアップ時も改訂したいルーチンの差し替えだけですみそうだし。個人的には階調の項目をもう少し説明してほしかったです。
藤原博人（25）X1turboZ 鳥取県

▼MZ-2500はもともと8ビットの中では突出している機能を持つマシンなのに，メーカーサイドからその力を生かせるようなグラフィックエディタが出ていないのは残念なことだが，画餅システムのようなサポートがあるのは素晴しいと思う。仕様もなかなか意欲的でこれからが楽しみだ。今後はウィンドウシステムやぼかしなど，読んでいるだけでも楽しいような部分も重点的にやってほしい。X1turboZや他機種のユーザーにも参考にできるような連載にしてもらいたい。また，「X68000マシン語プログラミング」で詳しくDB.Xを説明したのは正解といえるだろう。やはりマシン語のプログラミングにはデバッグが重要だ。今後は何か1本のプログラムを題材にして詳しく解説していってくれるとうれしい。それから，使用法が中心だった今月の「MZ-2500 MIDI入門」では，もう少し動作の一般的な解説もいれてくれたらもっと面白かった。また，「われら電脳遊戯民」が最終回になってとても寂しい。思うにゲーム界はいまだ甘えている。しょせんゲームだが，それでもゲームなのだ。オタクうけを狙ったグラフィック，型にはめて作ったような物語，なんら変化のない音楽，なにより，出ては消えていく数々のソフト。こうしたことは皆で考えなくてはならないと思う。
西田宗千佳（18）X1Fmodel 20，X68000 福井県

▼「DōGA・CGアニメーション講座」は，初回ということもあってかまだ「前置き」で今後どう展開していくのかははっきりしません。PCGAという点にはおおいに賛同します。そして，それを盛り上げるためソフトを柔軟に配布しようというやり方も必ず多くの賛同を得るでしょう。私見ですが，どちらかといえばアニメ的なCGのほうが興味があり，パソコンサンデーによれば2D用のツールも制作中とのことで期待しています。
湯沢聡（26）MZ-2500/2800，X1turbo，X68000，PC-1360K 東京都

▼いつも新しい視点からの考え方をみせてくれた「われら電脳遊戯民」が最終回になるとは残念です。今後もこのように斬新な切り口の記事をぜひ連載してください。
末吉克行（20）MZ-700，X1G/C，FM-7，MSX 兵庫県

ごめんなさいのコーナー

1988年9月号 DMACS
P.61 Y方向のドット数が8で割り切れない場合，画面下の一部が切れてしまいます。
A44E$_H$ 30 0A → 00 00
に修正してください。

バグに関するお問い合わせは
☎03(230)7683(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。
また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

いい汗ながそう Oh!Xin MIDSUMMER

▼特集のハードディスクとプリンタの活用編はいかがでしたか。フロッピーベースでは動作の難しいソフトも出てきてますます需要の高まっているハードディスク。ちょっとした仕組みを知って工夫すればいろいろ使えるプリンタ。それぞれ，パソコンの周辺機器の代表的なもののひとつとして皆さんの関心も高いことでしょう。これを機会に使いこなしに挑戦してみてください。今後の企画などに対する要望も含めて，みなさんのご意見をお待ちしています。

▼今月はページの都合により「(で)のショートプロぱーてぃ」はお休みです。ごめんなさい。でも，来月号ではマシン語ショートを予定しているから乞うご期待，と(で)氏がいっておりますので皆さん楽しみにお待ちください。

▼帰省先のある皆さんにお願いします。投稿作品などには，現住所といっしょに帰省しているときの連絡先も書いておいてください。作品に関してお尋ねしたいことがあるとき，「夏休みで田舎に帰ってるんだ」なんていう場合でも連絡できるとありがたいのです。どうかよろしく。

▼「残暑が続く長い夏になる」というのは気象庁の予測でしたが，今年は7月中からずいぶん台風の影響を受けましたね。こんなとき，長年あまり変わらないもののひとつとして，車のワイパーと並んで傘があげられると思います。「雨はなぜ空から降るんでしょう」「地面から降ってきたら傘がさせないから」なんてシャレにもなりませんが，確かに雨よけ用パーソナルツールとしては，現在の傘より優れたものは考えにくいですね。開くと内側に世界地図や星座が描かれているものもあったりして，う～ん世紀末にはこういうもんも流行るんだ，などとわけのわからない感心の仕方をしてしまいます。そうそう，かのE.T.だってPHONE HOMEするのにカサ使ってましたっけ。それをいうならメアリー・ポピンズさんは日傘で飛んできてたし，サザエさんにも「カサの効用」ってのがあった。

ますますわけがわからなくなりそうなのでまた来月。暑さ負けに気をつけて，夏休み後半も楽しくお過ごしください。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶1カ月ほどミシガンの大学生がホームステイしてたんだけど，これが豪気な奴で夜中まで人の部屋で遊んどいて翌朝「ねぼすけ」と起こしにくる。自分の時間が取れないので一計を案じ，居間にX1をおいてTETRISをやらせたら，そこから動かなくなった。助かった。
げに有り難きはロシア人の知恵かな。 (H.U.)

▶偉大なる指揮者ヘルベルト・フォン・カラヤン氏が逝去した。来年の2月に来日が決まった矢先の突然のこの出来事に，氏を愛するひとりとして大きな悲しみに包まれている今日この頃です。あーもう，試験も終わったしジメジメしててもしょうがない，夏なんだから。こうなったら海に山にと遊び回ってやるんだから。それ，レッツラゴー!! (H.K.)

▶すいません。ショートプロぱーてぃはいきなりその2（今月号）が休載ということになってしまいました。それから，先月号のその1では中見出しが落ちていました。そしてさらにイラストの高橋君の名前も落ちています。本当に皆さんにご迷惑をかけどうしで大変申し訳なく思います。来月号からはこのようなことがないように…するつもりです。 (で)

▶6月号の「組曲ユーフォリー」の感想を送ってくれた人，どうもありがとう。その中になんと原作曲者であるサコムの斎藤学氏からのがあったのには驚きました。そういえばサコムのPC88用新作「ヴァルナ」の曲も氏の作品ですが，これがまた，SB2対応のいい曲ばかりなので皆さん機会があったら是非聴いてみましょう！ (善)

▶パリ・コレももう秋冬。今年の話題はやはり,「フエレ」デザインのディオールだろう。新しいデザイナーの登用はシャネルのラガーフェルドなどの前例もあるが，今回はイタリア人デザイナーによるフランスブランドである。現状を変えていくのはたいへんなことではあるが，それを乗りこえなければ現状維持は(もちろん進歩も)ないのだ，と思う。 (C.W.)

▶ここに箱入りティッシュにかぶせるカバーがある。男の子が頬杖をついてうつ伏せに寝そべった格好をしてて背中からティッシュを取り出す。なかなか可愛いのだけど，背中がメキャッと裂けてズルズルと内容物をはみ出させる様は物体Xぽくもある。内臓模様のティッシュがあれば似合うかもしんない。今日はテレビで『The Thing』の新旧2本立て。(Mu)

▶「原子心母シジフォス」という名につられて，日触歌劇団の演劇を見に行ってしまった。その前に六本木でシンクロエネジャイザーしてきたせいかどうか知らないが，久々に面白い芝居だった。なんといっても客席まで飛ぶ血，棺からよみがえる吸血鬼が猟奇♡　不老不死の吸血鬼は，永遠に岩を押し続けるシーシュポスと同じ悲哀を背負っているのだ。(K)

▶無事アメリカ出張から戻ってくることができて嬉しい。アメリカといえば，最近アメリカの会社に転職した友人の言葉が忘れられない。粗挽きナントカ製法のジャ○ブという缶コーヒーがあるが，ジ○イブというのはスラングでオシ○コという意味があるそうだ。アメリカ人でこのコーヒーを買う人はいるのかなあ。 (伏せ字だらけのKO)

▶「恐怖は人間が現れるより以前から存在し，人間が滅びてからも存在するだろう」とは「幻の城」でバイロンを演じたときのヒュー・グラントの台詞。グラントの口から出たんでは，ナマ言うんじゃねーよ若造が，になってしまうのは惜しい。が，自分の創作した「恐怖」が実体化するという恐怖に捕らえられるマリー・シェリーはなかなかよかった。(よ)

▶ハードディスクの定価を調べる。あまりの実売価格との差に驚く。自民党の総裁候補について情報を得ようと『シアターアッパレ』を見る。載っていない……。洋行帰りのKO氏に話を聞く。「英語は通じたんですか」「C言語で筆談しました」なるほど。3カ月前に買ったはずのビデオデッキはまだ届かない……。 (U)

▶X68000の場合，次に望まれているゲームソフトというのは比較的ダイレクトに伝わってきやすいのだが，そのほかのアプリケーションについては皆さんからのリクエストの声が，最近あまり活発に聞かれなくなったような気がする。今後どのようなアプリケーションを期待しているのか，ご意見があればぜひお聞かせください。 (N)

▶写真で見ると左かなと思ったけど，動いていると やっぱり右だね。発売日に買ったLD（3曲入りで2400円）の感想を話すと「まったくオジサンなんだからぁ」なんて声が返ってきた。ふふふ，どちらがいいって言えるのはまだだいじょうぶ。区別がつかないようだと完全にオジサンだよ。とはいえ，一回りも違うんじゃ物理的にオジサンか。 (T)

microOdyssey

最近，「Dyna Book（J-3100SS）ってどうなんでしょうか？」と聞かれることが多い。

私のまわりにはそれ以前からポータブルワープロをほしがっている人が結構多かった。本当に持ち運べて，年賀状以外の用途にもちゃんと使いものになるものを，だ。ポータブルワープロといっても，どこがポータブルなのかよくわからない重いマシン（ああ，PC-1600K はポータブルコンピュータだったのに）はあったが，速度も原稿書きなどをするには耐えられないくらい遅いものがほとんどだった。

そのうち，大きさ的には手頃なものが現れた。いわゆるノートワープロだ。ノートワープロ各機種を見渡したなかではSONYのPRODUCE 1000がもっとも理想に近いものだった。が，実際に触ってみるといまひとつ使い勝手が悪く，また，機能を見ていると「ここまでやるなら，なんでパソコンにしないんだ」という苛立ちを覚えてくる。これはほかのノートワープロに対しても感じることだ。パソコンなら気にいらない部分に手を加えて……といったことも考えられるが，押し着せの機能しか持たない専用機ではどうしようもない。

こうなれば，増設プリンタ部の中身をくり抜いて1ボードコンピュータを入れ，RS-232Cで本体と接続する，とか遊び半分ながら，結構真面目に改造計画を練ったこともあった。それほどまでに，持ち運べるワープロを渇望していたのだ（もっとも，自由時間を削って仕事できるというだけのメリットしかないのだが）。

そんなおりに突然発表されたのがこのDyna Bookだ。パソコンならワープロにもなるだろう，という期待もできるし，スペックがそれまでのポータブルパソコンを遙かに凌いでいる。期待するなというほうが無理なのだ。

問題がないわけでもない。透過型液晶パネルを使うとバックライトの分だけ電力消費が多くなり，反射型液晶に比べて動作時間が短くなるといった指摘もある。従来タイプのJ-3100SLでは反射型液晶で連続使用4時間が可能であったのだが，J-3100SSではそれが2.5時間に短縮されてしまったのだ。そのほかJ-3100シリーズの周辺機器が接続できないなども，やや気にかかる。今後の各社から予想される対抗機種によって，決定版とはならないかもしれない。

しかし，それでもなお，このマシンの果たした役割り，果たすべき役割りは大きい。小さなデスクトップとしてしか使われないラップトップでは意味がない。パソコンユーザーの多くが願っていたマシンが現実に近づいたという意味で素直に喜びを感じる今日この頃である。

東芝のJ-3100シリーズは世界でいちばん売れているラップトップマシン。高級機では，80386に100Mバイトハードディスクを積んでUNIXが動く。世界的にも評価は高い。しかし，単なるラップトップMS-DOSマシンに“Dyna Book”と名づけるならば，486でも載せた32ビットUNIXマシンには“HAL9000”という名前をつけかねないな……と思うと妙に恥ずかしくなる。もし，J-3100SSが富士XEROXにOEM供給されてXEROXブランドの“Dyna Book”が発売されることにでもなったら（決してありえない話じゃない），アラン・ケイはいったいどんな顔をするだろうか。　（U）

1989年10月号9月18日（月）発売

特集　ゲーム面白心理学

最新ゲーム要素徹底分析　ファンタジーゾーン/ダブルイーグル/サバッシュ/ねじ式/維新の嵐　他

ゲーマー必見!!　X68000ゲーム年表

MZ-700System-7B用シューティングゲームSIDEROLL-F

X1用カードゲーム　Bonding

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートB1	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(257)2660
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊国屋書店本店	03(354)0131
	高田馬場	未来堂書店	03(200)9185
	渋谷	大盛堂書店	03(463)0511
	池袋	リブロ池袋店	03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は，とじ込みの振替用紙の「申込書」欄に何年何月号からをご記入のうえ，**年間購読料6,720円（税込）**を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!X定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

Oh!X　9月号

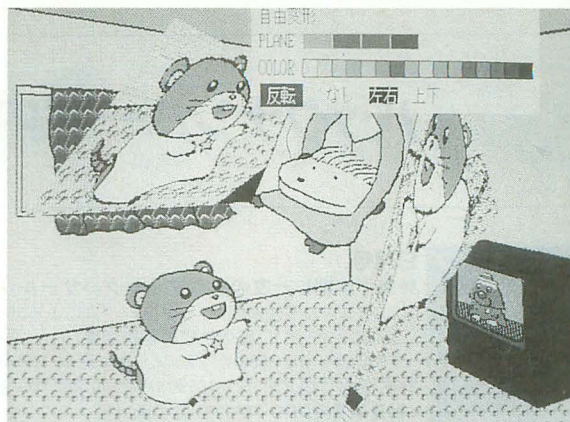
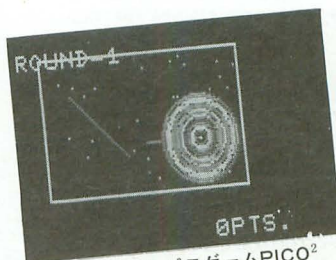
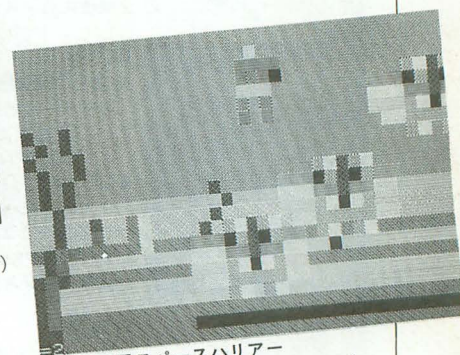
■1989年9月1日発行　定価560円（本体544円）
■発行人　孫　正義
■編集人　橋本五郎
■発売元　（株）日本ソフトバンク
■出版事業部　〒102 東京都千代田区九段南2-3-26　井関ビル
Oh!X編集部　☎03(230)7681
出版営業部　☎03(230)7670　FAX 03(262)8397
広告営業部　☎03(230)7672
■印　刷　凸版印刷株式会社

投稿プログラム大募集

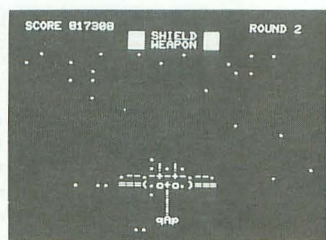
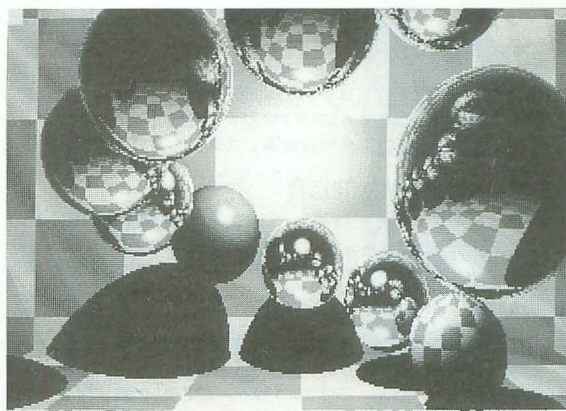
のお知らせ

Oh!Xでは，毎月さまざまな投稿プログラムを掲載しております。これらはすべて，ゲーム音楽を聞いているうちに自分のマシンで演奏してみたくなった，市販のものもあるけどもっと便利なグラフィックツールが欲しかった，またはMZ-700でスペースハリアーを遊びたいなど，どれも皆さんが日常のなかでパソコンと接しているうちに，ふと思いついたことを形にしようと努力して生み出された傑作，名作ばかりなのです。

でも，読者の皆さんがそうして作り上げたプログラムを，一部の方を除いては自分のディスクのなかだけにしまっておくのはもったいない話。ひとりでも多くのユーザーに使ってもらえば，またそれをベースにして新しいプログラムが生まれる可能性だって広がるのです。

ですから，Oh!Xではそういったちょっとしたきっかけを機に，完成度の高いものよりも自分のアイデアをそのまま形にしたような，オリジナリティあふれる投稿プログラムをスペースを空けてお待ちしています。もちろん，ピコピコゲームのようなショートプログラムも大歓迎。自信作をお持ちの方は，募集要項をよくお読みのうえぜひご参加ください。お待ちしています。

MZ-2500用グラフィックツールDMACS（1988年9月号）

MZ-2500用ピコピコゲームPICO^2（1988年4月号）

MZ-700用スペースハリアー（1988年10月号）

X1/X1turbo用割り込みミュージックシステムPSI（1988年3月号）

X68000用ストラテジーゲームSTAR TREK（1988年11月号）

S-OS"SWORD"用ELFES（1988年2月号）

X1turbo用レイトレーシングツールturbo RAY TRACER（1988年9月号）

投稿募集要項

1） お送りいただくプログラムには，住所・氏名・年齢・職業・連絡先電話番号・機種名・使用言語・必要な周辺機器・マイコン歴等を明記のうえ，封書の宛て先の最後には「Oh!X LIVE」や「S-OS"SWORD"」，「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2） 投稿されるプログラムには，詳しい内容を記入した原稿と一緒にフローチャート，変数表，メモリマップ，参考文献などの資料もお書き添えのうえお送りください。また，お送りいただいた原稿については，当方で加筆，修正させていただく場合があります。

3） お送りいただくプログラムは最低2回はセーブしてください。基本的に同封されたカセットテープおよびフロッピーディスクについてはご返送いたしませんので，あらかじめご了承ください。

4） ハード製作関係の投稿につきましては，最初は詳しい内容のわかる原稿のみお送りいただければ結構です。その後，当方において製作物が必要だと判断した場合は，改めてご連絡いたします。

5） お送りいただいた投稿プログラムの採用につきましては，掲載月号が決定した時点で当方よりご連絡を差し上げます。特に各種ツール関係，ハード関係のものにつきましては，特集内容などを考慮したうえで採用が決定されることがありますので，採用結果をご連絡するまでに時間がかかってしまう場合もあります。

6） 投稿いただいたプログラムにバグ等が発見された場合には，新しいプログラムの入ったメディアと一緒に，文書にてご連絡ください。

7） 掲載された投稿プログラムに対しては当社規定の原稿料をお支払いいたします。また，プログラムの著作権等は制作された方に保留されますが，PDSとしてネットなどにアップロードされる場合は，必ず編集室まで事前にご連絡ください。なお，一般的モラルとして，他誌との二重投稿または，他誌に掲載されたプログラムの移植などについては固くお断わりいたします。

宛て先
〒102 東京都千代田区九段南2-3-26 井関ビル
日本ソフトバンク Oh!X編集室「投稿プログラム」係

BACK ISSUES

バックナンバー案内

ここには1988年9月号から1989年8月号までをご紹介しました。現在1987年4, 1988年1, 2, 4～12, 1989年1～8月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については本文160ページを参照してください。

1988

9月号

特集　半期に一度のグラフィックバザール

CGアニメの手法入門/ワイヤフレームによる3D/X68000スプライト/画像処理の基礎知識/turbo RAY TRACER/MZ-2500用グラフィックエディタDMACS

THE SOFTOUCH C-TRACE68/SAMPLING PRO-68K 他

C調言語講座PRO-68K(3) 謎の低次元グラフィック

MIDI活用テクニック(2) 割り込みによるMIDI通信

Z80マシン語ゲーム工房(2) 応用への基礎固め

全機種共通システム ラインエディタTED-750/WINERの拡張

10月号

特集　百花繚乱ゲームバトルロイヤル

最新ゲーム総登場　ハイドライド3/A列車で行こうII/たんば/熱血高校ドッジボール部/フルスロットル　他

MZ-700用SPACE HARRIER

- Oh!X LIVE 1974(16光年の訪問者)/瑠璃色の地球/二人のゼネレーション/バッハのアリア

MIDI活用テクニック(3) 複数の音源を操るテクニック

C調言語講座PRO-68K(4)/Z80マシン語ゲーム工房(3)

全機種共通システム SLANG用拡張ライブラリ/MANKAI

11月号

特集　いまどきのプリンタ活用術

メカニズムを理解しよう/制御コード/文字と図形の混在印字/拡大文字のスムージング/外字登録ツール/S-H COPY/グラフィックのモノクロ出力/X68000のCOPYキー/オリジナル印刷キット/試用レポート

THE SOFTOUCH NEW Print Shop PRO-68K 他

OS-9/X68000入門(1) OS-9ってなに？

- STAR TREK for X68000

全機種共通システム シューティングゲームELFES IV

12月号

特集　パソコンはいま音楽の領域へ

なぜ自動作曲か/心地よい雑音の話/和音の読み方/美しい響きの要素/4分音符は歌い始める/古くて新しい音楽形式/FM音源の仕組み/Melody Box/MusicBASIC

- さよなら Live in '88　バッハ イタリア組曲他6本
- Oh!X　1周年記念特別企画「ちょっとあぶない福袋」

OS-9/X68000入門(2) OS-9のオペレーション環境

Z80マシン語ゲーム工房/C調言語講座PRO-68K

全機種共通システム ソースジェネレータSOURCERY

1989

1月号

特集　いきなり初春からハードウェア

デジタル回路入門/電子サイコロ/乱数発生器/X1turboバンクメモリ拡張/X68000用CP/M-80システム　他

1988年度GAME OF THE YEAR ノミネート作品発表

- MZ-2500用　Hyper Game Book
- LIVE in'89　エンデューローレーサー/アルルの女
- ようこそ，セガ・メガドライブ!!

C調言語講座PRO-68K/Z80マシン語ゲーム工房

全機種共通システム パズルゲーム LAST ONE/FLICK

2月号

特集　マシン語“でじたるざんまい”

アーキテクチャからのマシン語入門/アセンブラへの招待/超入門Z80マシン語活用術/X68000料理教室

THE SOFTOUCH 彩CRONE/Final Ver.3.2　他

- X1/X1turbo用RPG FLAME

Z80マシン語ゲーム工房　最終回 爆発，そして完成へ

C調言語講座PRO-68K (8) とおりゃんせなのである

OS-9/X68000入門(3) ついに発売！OS-9/X68000

全機種共通システム 高速エディタアセンブラREDA

3月号

特集　BASIC“おもちゃ箱”

ピコピコゲームから重力シミュレーションまで

- X1/X1turboでMZ-700用スペハリ/ロボットゲームTAMA
- 数値演算を高速化　FLOAT2＋.X

OS-9/X68000入門(4) C言語の概要を見る

C調言語講座PRO-68K(9) ニホン語，不得意

新連載予告編X68000マシン語プログラミング入門

全機種共通システム 浮動小数点演算パッケージSOROBAN

THE SOFTOUCH/LIVE in'89/知能機械概論/猫とコンピュータ

4月号

特集　ゲーマーたちの“新深夜族”宣言

1988年度GAME OF THE YEAR

新連載　X68000マシン語プログラミング

- X1/turboパズルゲーム　ロボット衛兵
- MZ-700用ゲームパッケージ System-7B
- LIVE グラディウスII/ザ・スキーム/パワードリフト

連載　C調言語講座PRO-68K/OS-9/X68000入門

全機種共通システム　SLANG用実数演算ライブラリ

特別付録　X68000イメージCGポスター

5月号

特集　MIDIサウンドデータ料理術

LA音源をFM音源でシミュレート/X-BASICでMIDI制御

特別企画　第4回「言わせてくれなくちゃだワ」

- シャープパソコンフォーラム'89 in赤坂
- 詳解Human68k ver.2.0
- MZ-2500，X1/X1turbo用　戦略的ライトサイクルゲーム

連載　C調言語講座PRO-68K/OS-9/X68000入門
X68000マシン語プログラミング

全機種共通システム　ソースジェネレータRING

6月号

特集　これからのXfamily

X68000に光磁気ディスクを/学習リモコンの製作

THE SOFTOUCH ライトニングバッカス/Might and MagicII他

- OPMA用外部関数による　KENBAN.BAS
- X1/X1turbo用ドライブゲーム　Spirit of Rally
- X1turboZ用　これ，パズルなんですか。

MZ-2500 MIDI入門(1) MIDIボードを作る

C調言語講座PRO-68K/X68000マシン語プログラミング

全機種共通システム　超小型コンパイラTTC

7月号

特集　3Dグラフィックへの飛翔

Zバッファアルゴリズム/スムースシェイディング　他

THE SOFTOUCH Terazzo PRO-68K/アドヴァンスト・ファンタジアン

新連載
- DōGA・CGアニメーション講座
- MZ-2500用グラフィックエディタ作成講座
- マシン語カクテル in Z80's Bar
- X-BASICプログラミング調理実習

全機種共通システム TTC用パズルゲームTIC BAN

X68000マシン語プログラミング/C調言語講座PRO-68K　他

8月号

特集1　X1プログラミングガイドブック

PCGの基礎から奥義まで/超高速ラインルーチン　他

特集2　3Dグラフィックの深淵へ

スキャンラインZバッファ/3Dモデリング　他

新連載 (で)のショートプロぱーてぃ

X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/DōGA・CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar　他

全機種共通システム　CP/M用ファイルコンバータ

好評既刊

Oh!MZ1987年7月号まで25回にわたり連載されたユニークなエッセイが，加筆・修正のうえ再編集されて一冊の本になりました。パソコン好きのダンナ様と一人息子，それに，ときどき人間よりも人間らしい白猫ホンニャアが，著者の筆先から生き生きと動き回ります。扉を開けたら，そこはもう“たかざわきょうこの世界”。きっとあなたも，猫かコンピュータがほしくなることでしょう。

A5判 定価1,200円(税別)
高沢恭子 著

猫とコンピュータ

BOOKS

好評既刊

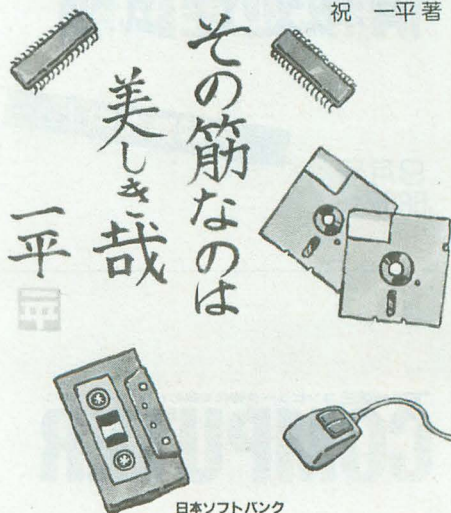
内容

特別付録 X1処理技術者試験

X1のハードウェアをくまなく探検した祝一平氏の名著。オリジナルプログラムも豊富に掲載。ユーザー必携です。

試験に出るX1
ハードウェアのフルコース

祝 一平 著

B5判 定価2,800円(税別)

株式会社 日本ソフトバンク出版事業部
〒102 東京都千代田区九段南2-3-26 ☎03(230)7670

日本ソフトバンクの
書籍特約書店

下記の書店の一覧は、日本ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。
(注)現品が売れて補充中の場合もございますので、ご注意下さい。

SOFT BANK

日本ソフトバンク出版事業部

〒102 東京都千代田区九段南2-3-26 ☎03(261)4095

全国特約書店一覧

地域	書店名	電話
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善札幌支店	011-241-7252
〃	リーブルなにわ	011-221-3800
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東　北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	ブックイン城東	0172-28-2882
八戸市	伊吉書院	0178-44-1917
盛岡市	東山堂書店本店	0196-53-6464
〃	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	高山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
〈関　東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店ホリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	0488-22-5321
浦和市	須原屋コルソ店	0488-24-5321
大宮市	押田謙文堂	0486-41-3141
〃	ブックセンター押田	0486-47-3141
〃	三省堂ブックポート	0486-46-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	0487-52-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	巌翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二巌翠堂	0474-65-0926
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	ブックセンター文教堂	044-811-5557
〃	文教堂溝ノ口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462·23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本店	0427-74-5581
〃	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
〈東　京〉		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	巌翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂ＮＳ店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂ＮＳビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リブロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	リブロ錦糸町店	03-846-0111
〃	ブックストア・談	03-635-1841
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229

展示図書一覧

MS-DOSいたれりつくせり本 ●1800円
プレイMS-DOS ●1900円
UNIX System V プログラマ・ガイド ●12000円
UNIX System V ユーザ・ガイド ●9800円
UNIXオペレーティングガイド ●3000円
C言語の活用理解 ●2000円
C言語の基礎知識 ●2500円
C言語の応用50例 ●2300円
上級・C言語の応用例50例 ●2400円
Cプリプロセッサ・パワー ●2200円
Play the C 上巻 ●1500円
Play the C 下巻 ●1500円
Turbo C入門 ●2600円
8086アセンブリ言語 ●2800円
8086マクロプログラミング ●2600円
ビギニングMUMPS ●2600円
Final Ver.4.0ブック ●2400円
MIFES Ver.4.0ブック ●2400円
ビジネスソフトデータ活用ブック ●2800円

BASICによるプログラミングスタイルブック ●1800円
ソーティング・ノート ●1900円
BASICプログラムジェネレータ集 ●2800円
98/88スモールビジネスプログラム集 ●2500円
88デスクアクセサリ集 ●2000円
J-3100パワーユーザーブック ●2400円
フロッピーディスクフル活用ガイド ●2300円
PC工作入門 ●1800円
続・PC工作入門 ●1800円
PC-286Lブック ●1700円
試験に出るX1 ●2800円
Lotus1-2-3ガイドII ●2500円
MS-Chart Ver.3.1ガイド ●2900円
まいと〜くガイド ●2300円
新松ガイド ●2000円
一太郎Ver.3ガイド ●2500円
新一太郎ガイド ●2300円

桐Ver.2ガイド ●2500円
花子応用ガイド ●2500円
Lotus 1-2-3ガイド ●2400円
P1ガイド ●2300円
Ninja2ガイド ●2300円
Multiplan Ver.3.1ガイド ●2400円
アセンブラCASL入門 ●2000円
ハードウェア徹底マスター ●2500円
FORTRAN徹底マスター ●2800円
情報処理の基礎知識 ●1600円
COBOL徹底マスター ●2900円
受験用語ハンドブック ●1800円
ワープロ文書F・O・P ●1200円
バイト&ワードの風にのって ●1800円
田原総一朗のパソコンウォーズ ●1400円
コミック・トロン革命 ●1200円
ムーグ・ノイマン・バッハ ●1300円
RPG幻想事典 ●1500円
RPG幻想事典・日本編 ●1800円
魔法王国シムルグント ●1800円

市	書店	電話
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	久美堂東急ハンズ店	0427-28-2772
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
上田市	平安堂上田店	0268-22-4545
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店SBS店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
〃	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
名古屋市	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
一宮市	三洋堂一宮店	0586-77-5734
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
〃	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
山口市	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	029-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	0963-22-5531
〃	長崎書店	0963-53-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

日本ソフトバンクの電脳本（コンピュータブックス）

C言語

●Quick Cの本格活用を示す
QuickCプログラミング
エディットからコンパイル・実行までの統合環境を備えた「QuickC」の本格的入門書。
野口雄平・池田和徳・中島昌二 共著　☆定価2,680円

●C STEP UPシリーズ①
上級・C言語の応用50例
画面制御編、ディスクドライブ制御編ほか大きく4つに分け、それぞれくふうを凝らしてプログラムを作成。
上原・中山・石田・井上 共著　定価2,400円

●C STEP UPシリーズ②
C++プログラミング
今話題の言語、C++の本格的入門書。本書の各項目をカード別にまとめ、サンプルリストや図版を豊富に掲載。
門内淳・赤堀一郎 共著　定価2,600円

●BASICからのアプローチ
C言語の活用理解
具体例を数多く取り上げ、BASICの知識を有効に利用しつつ、BASICからCへ順次移行できるように解説。
保坂光・若林淳 共著　定価2,000円

●基本からのアプローチ
C言語の基礎知識
短いけれど実用的なサンプルプログラムを紹介しながら、C言語の基礎知識を解説しています。
門内淳・乗松保智・上原哲郎 共著　定価2,500円

●実践からのアプローチ
C言語の応用50例
C言語を使い98上でどのようなプログラムが作れるのかを、豊富なサンプル例によって示しています。
上原・石田・乗松・中山・高木 共著　定価2,300円

●初級C言語講座
Play the C[上・下]
Cを身につけたい、これまでの解説書ではもの足りない、そんな要求に応えCの世界を系統的に捉え直しています。
林 晴比古 著　定価各1,500円

●初めてのTurbo Cプログラミング
Turbo C入門
初心者にとってなじみのうすいTurbo C開発環境をたくみに解説しながら、具体的で示唆に富んだ用法をも満載した決定書。
林 晴比古 著　定価2,600円

●C言語の秘められた能力を解き放つ
Cプリプロセッサ・パワー
林 晴比古 著　定価2,200円

OS

●APIごとにCのサンプルプログラム満載
OS/2APIブックⅠ カーネルAPI編
OS/2プログラミングユーザーのための本格リファレンスブック。APIごとの図とCによるサンプルプログラムを豊富に掲載。
坂本 浩 著　☆定価2,790円

●ビジネスソフトユーザーのためのOS入門
MS-DOSいたれりつくせり本
アプリケーションソフトユーザーが効果的な〝ファイル整理〟をするためのMS-DOS入門書。
佐藤みどり 著　定価1,800円

●ビギナーのためのレクチャーブック
プレイMS-DOS
MS-DOSを学ぶうえで必要な知識と必要でない知識とをふりわけ、初めての人にでも無理なく学べる入門書。
林 晴比古 著　定価1,900円

●はじめてのシェルプログラミングからUNIX応用まで
UNIXオペレーティングガイド
UNIXの基礎知識から、シェルプログラミング、MACSYMA、Sについて豊富な例を用いながら紹介した入門書。
渡辺・岸本・佐藤 共著　定価3,000円

UNIX System V プログラマ・ガイド リリース3.1
最新バージョンRel.3.1上で動作する言語、デバッグ、オブジェクトの操作等に関連するユーティリティ、データ構造について解説。
定価12,000円

UNIX System V ユーザ・ガイド リリース3.1
UNIX System環境の要素であるシェルやvi、ed等のエディタ及び電子メールなど各種ユーティリティについて、詳細に解説。
定価9,800円

アセンブラ

●マシン語プログラミングの基礎から応用まで
8086アセンブリ言語
8086の各命令に始まり、アセンブラの使い方から、GDCのプログラミングまでを解説した格好のマシン語入門書。
西村義孝 著　定価2,800円

●MACRO86(MASM)活用テクニック
8086マクロプログラミング
MS-DOS上で動作するアセンブラMACRO86に焦点をあて、その使いこなし方、デバッグの方法を詳細に解説。
西村義孝 著　定価2,600円

☆印は税込み、無印は税抜きの定価です。

SOFT BANK 日本ソフトバンク出版事業部

SPECIAL ISSUE OF KNOWLEDGE★INDUSTRIAL COMPANIES★PUBLISHED BY NIHON SOFTBANK

コンピュータが好きな大学生のためのバイブルブック

THE ソフトバンク '89

THE SOFTBANK

マジメでアップ・ツー・デートな話題を満載

純生

月刊情報処理試験5月号別冊

定価500円(本体485円)

就職読本 ザ・ソフトバンク

見セマス。コンピュータの全て。最新情報。

コンピュータの歴史はわずか50年。コンピュータの一年には他の産業の10年が凝縮されている。「そもそもコンピュータとは何か」から始まり、「これからどうなる」までわかるハズ。

見セマス。ザ・就職。

ラスト一年。君はどう戦う？ 解禁までのすごし方。訪問。そして……。どんな職種があるの？ 君に向いてる分野は？ これからの君が見えるハズ。

見セマス。企業。

バンバン手元に届く企業情報。どれをどう目を通すか？ このギョーカイの会社を辞書にしました。この一冊が君の会社研究のバイブルになるハズ。

★全国一斉好評発売中!!
お近くの書店・大学生協でお求めください。

大特集！見せます、コンピュータの現在・過去・未来

THE TOPICS 15

ソフトウェア業界にもいろんな商売があるんだ

保存版 業界就職へのCheck Point

メーカ別に見る"広告戦略"あれこれ

生涯教育・再教育

これがソフトウェア開発だ

就職読本 ザ・ソフトバンク

定価500円

無料贈呈■ザ・ソフトバンク'89をご希望の方は電話でお申し込みください。

TEL(03)235-9379

ザ・ソフトバンク'89掲載に関するお問い合わせも受付けています。 TEL(03)235-9379

株式会社 日本ソフトバンク
新規事業企画室分室
〒162 東京都新宿区市谷田町2-37千代田ビル

いろんな分野から業界をみせる

情報・ソフトウェア まさにこれからの基幹産業。メーカー系とユーザー系と独立系に分かれ、ソフトウェア開発を中心に行なっています。

メーカー 大型のメインフレーマからパソコンまで、コンピュータのハードウェア本体を製作する企業

商社 ハードとソフトを流通させるキーポイント。取扱うモノが技術製品であるだけに、専門知識もかなり必要とされます。

百貨店・ストア・専門店 エンドユーザーに直接製品を販売するだけでなく、VAN事業に取り組むなど活躍の場を広げています。

金融・証券・保険 オンラインの国際化にともない、現在かなり大規模なコンピュータ化が進められています。今後も需要の多い分野。

その他 コンピュータの保守サービスや、異業種からの参入など、上記の5種類に含まれない企業群です。

この他にも内容は盛りだくさん。たとえば、某企業がクルマの流線型フォルム造りを手がけている話とか、パソコンをファクシミリとしても使用可能にしたソフトウェア会社の話など。いろんな企業から教えてもらう、知れば必ず得する情報をギッシリ掲載しています。

これで、〝おタク族〟の先輩をやめてください。
友だちに「おタク」と呼びかけるパソコン少年たちは、〝おタク族〟とも言われています。この言葉からイメージするのは、専門知識だけで頭でっかちになり、キーボードとCRTディスプレイしか視野にない子どもたち。私たち『日本ソフトバンク』は、コンピュータ好きな学生達に、そんな〝おタク族〟の先輩でいて欲しくありません。専門知識はもちろん、視野を広く業界に拡げて、情報通になっていただきたいのです。そのための情報誌「ザ・ソフトバンク」。最新の話題を満載しています。ぜひ、ご一読ください。
無料だぜ！
THE ソフトバンク 6
THE・SOFTBANK
コンピュータ関連企業ウオッチング
TSB REPORT
次は8月号だ！
来春卒業する大学・専門学校の諸君に無料で送る
ザ・ソフトバンク'89 6月号

電脳作家発売1周年記念作品！ 9月発売予定
本格的ファンタジーアドベンチャーゲーム

AQUARIUS－宝瓶宮－

対応機種：X68000(5インチ2HD)4枚組　**¥3,980**
シナリオ：神戸大学情報統計部　赤坂賢洋
グラフィック：神戸大学情報統計部　細見 格・中野博之

アクエリアスは、電脳作家(サイバーライター)Ver2.0上で動作する全4部からなる超大作シナリオソフトです。美しいグラフィックとBGMをふんだんに使った本格的ファンタジーアドベンチャーゲームをお楽しみいただけます。

●あらすじ
強く賢い少年オルフェウス、類希なる美少年ガニメデ、優しく美しい少女アストレアという大変仲の良い3人トリオがいた。歳月が流れるうち、やがて、オルフェウスとガニメデは、共にアストレアを愛するようになった。アストレアは、悩んだ末、オルフェウスを選んでしまった。怒ったガニメデは、魔女ヒュードラに自らの魂を代償にオルフェウスに災いをなすように頼んだ。ヒュードラは、若く美しいアストレアに嫉妬するあまり、ガニメデの願いを聞かず、アストレアをさらって、魔法で石に変え、洞窟に閉じ込めてしまったのであった。オルフェウスは、アストレアを助け出すため、一人旅立つのであった……

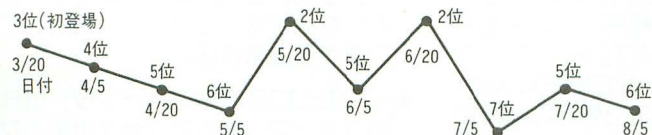
10月発売予定
英語の予習時間70%カットに成功！辞書4000語付き！
翻訳ヘルパーずるかまし ¥5,980

対応機種：X68000(5インチ2HD)2枚組
開発者：大阪市立大学マイコン研究会　山本 賢一(プログラム)
山本 博之(辞書)

驚異のロングビッグセラーシューティングゲーム！

D-RETURN

対応機種：X68000 (5インチ2HD)2枚組　**¥5,980**
開発者：神戸大学情報統計部 前部長　赤坂 賢洋

日本ソフトバンク出荷本数によるランキング推移
「Oh!新作売れ筋」掲載のX68000ソフト中でのD-RETURN(3月1日発売)ランキング

第2回 D-RETURN ハイスコアコンテスト結果

	得点	氏名	年令	居住地
1.	22,553,390	森沢建行	17才	高知
2.	7,381,330	井谷　隆	19才	兵庫
3.	6,291,346	上鍵忠志	17才	大阪
4.	5,462,660	永田秀和	17才	愛知
5.	4,351,350	上農友寛	16才	大阪

上位5名様を日コン連特別技能会員に認定。
第3回締切は、9月30日消印有効。

郵送品貼付切手には、オール記念切手使用！
日コン連SOFT通信販売のご案内
現金書留、郵便振替(大阪5-4873日コン連企画株式会社)、為替、定額小為替で、希望商品名、対応機種名、数量明記の上、お申し込みください。(送料はサービス。)
このうち、現金書留、定額小為替でお申し込みの場合には、例えば、5,980円の商品の場合には、端数を切り上げ 6,000円分お送り戴いて結構です。この際のおつり20円は、商品発送時に同額の記念切手でお返しいたします。

AN ADVENTURE GAME INTERPRETER
Cyber Writer
電脳作家 Ver 2.0

対応機種：X68000(5インチ2HD)2枚組
開発者：神戸大学情報統計部　部長　村尾 元　**¥5,980**

電脳作家は、専用の言語で書かれたシナリオをX68000上で、コマンド選択式アドベンチャーゲームの形で実行する一種のインタプリタです。Ver2.0では、OPMによる音楽演奏やPCMによる音声出力も可能となり、より良質のアドベンチャーゲームが作れるようになりました。便利なグラフィックツールに加え、買ったその日から遊べるサンプルシナリオ付きです。

電脳作家グラフィック & ミュージックライブラリー集
制作者：神戸大学情報統計部　細見 格・赤坂賢洋　**¥3,980**
◆グラフィックデータ10ファイル、ミュージックデータ39ファイル収録。

シナリオコンテスト入賞作品通信販売中！
EVIL EYE 作：三上潤一郎(Jun.M.Win) Ver2.0対応
¥1,000(日コン連企画㈱まで、直接お申し込みください。)
内容：えいぶるという少年が、ある世界に平和を取り戻すために旅するというもの。
特長：BGM、音声出力をフルに使ったユニークなアドベンチャーゲーム。

お知らせ！

日本コンピュータクラブ連盟加盟団体募集中！
加盟費、会費一切不要！

●日コン連関東本部紹介
《関東本部加盟団体》
横浜市立大学パソコンクラブ、東京学芸大学教育工学研究会、東京水産大学コンピュータクラブ、青山学院大学MEET COUNT、早稲田大学パソコン同好会、成蹊大学FAC電子計算機研究会、工学院大学電子技術研究部、NEW MZM、MAC、Peke、EXTEND PARTY
大学サークル、同人ソフト団体 大歓迎！ 関東地区加盟希望団体は、日コン連関東本部委員長、TEL 03-407-4130 後谷行則まで。

●日コン連全国本部付スタッフ募集(高校生・大学生歓迎)
条件／日コン連全国本部(なんば)に頻繁に来られる人。
特典／日コン連全国本部にある各パソコン、ソフト使い放題。
仕事内容／各パソコンメーカーとの交渉、日コン連加盟団体への連絡、イベント企画・立案。11月創刊の日コン連発行パソコン雑誌の編集協力。

●日コン連発行のパソコン雑誌ライター募集
京阪神地区在住の方で、パソコンやソフトについての情報、意見、解説、評価ならびにショップ情報、タウン情報を提供していただける方を募集します。大阪制作初のパソコン雑誌創刊にご協力ください。

●日コン連パナコムリクルーティングSIG開催
9月30日まで、上新電機J&P HOTLINE上で開催。
電子メールを使って人事採用担当者に氏名・身元を明かさずに直接、集問することができます。
情報サービス業クイック社に興味のある方は、お気軽にご利用ください。

日コン連SOFT保証
日コン連SOFTのディスク内容をお客様が破損された場合、そのディスクと300円分の切手を同封してお送り頂ければ、折り返し、新しいディスクをお送りしています。

●問い合せ・申し込み先
日コン連 SOFT
〒556 大阪市浪速区難波中2-4-3 村上ビル
TEL 06(644)6901
日コン連企画株式会社または日本コンピュータクラブ連盟

信用と実績を誇る BASIC HOUSE

北関東最大の68000専門店

SHARP SONY NEC YHP 横河ヒューレットパッカード Apple Computer

BASIC HOUSEで68000CPUが大流行

こだわり派向けのマンハッタンシェイプ X68000 EXPERT

CZ-602C&CZ-602D	CZ-612C&CZ-612D
¥390,000(税込み)	¥490,000(税込み)
36回分割	36回分割
第1回 ¥13,400	第1回 ¥18,900
第2回 ¥13,100×35回	第2回 ¥16,400×35回

スロットいっぱいローコストタイプ X68000 PRO

CZ-652C&CZ-603D	CZ-662C&CZ-612D
¥320,000(税込み)	¥440,000(税込み)
36回分割	36回分割
第1回 ¥12,700	第1回 ¥17,900
第2回 ¥10,700×35回	第2回 ¥14,700×35回

X68000用ハードディスク

Logitec	Logitec	アイテック	SHARP
LHD-34V	LHD-32V	ITX-203	CZ-620H
~~¥158,000~~➡ ¥134,000	~~¥128,000~~➡ ¥110,000	~~¥125,000~~➡ ¥106,000	~~¥178,000~~➡ ¥ 89,800
容量40Mバイト	容量20Mバイト	容量20Mバイト	容量20Mバイト
オートシッピング機能 PC98シリーズ用ですがX68000でも動作します	オートシッピング機能 PC98シリーズ用ですがX68000でも動作します		メーカー純正 特別奉仕価格

BASIC HOUSE X68000 オリジナルソフトウェア

B6-6301	B6-6302	B6-6303	B6-6304	B6-6305	B6-6306	B6-6307
BASIC 拡張関数パッケージ ¥9,800	CP/M68K エミュレータ ¥19,800	アイコンエディタ ¥4,800	ディスクキャッシャー ¥6,800	C言語ライブラリー ¥6,800	BASIC 拡張関数パッケージ (C言語ライブラリ付) ¥14,800	Toys & Tods ¥6,800
X-BASICの機能をアップさせる約50種の関数パッケージ	CP/M68KのBDOSコール機能をエミュレートし、CP/M68KのアプリケーションをHuman68Kで実行します。	ビジュアルシェルで使用するアイコンを登録変更します。	Human68K用ディスクキャッシュドライバ	BASIC拡張関数パッケージをX BASTOCで利用するためのC言語ライブラリー	B6-6301とB6-6305のセットです。	Human68Kで使用する外部コマンドをセットにしたものです。

New BASIC House Open!!

Basic Houseに大田原営業所ができました。

美原公園野球場裏、ダイユー敷地内
白い建物です。よろしくお願いします。

TEL.0287-23-5352

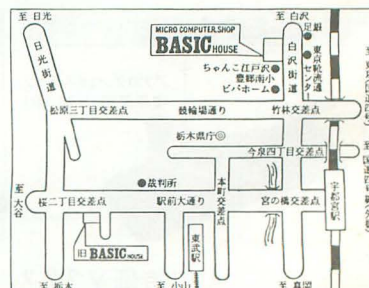
大田原営業所では通信販売は扱っておりません。
通販ご希望の方は宇都宮本店にお願いします。

通販御希望の方は購入品名、住所、氏名、電話番号を書いた紙と(代金＋送料¥1,000)消費税1.03を同封して現金書留でお申し込み下さい。釣銭は無いようお願いします。

全国通販OK!

- ●低金利クレジットあつかっております。
- ●支払方法は相談に応じます。

表示価格は特に明記されている場合を除き消費税は含まれておりません。

MICRO COMPUTER.SHOP
BASIC HOUSE

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社**計測技研**　本社営業部／マイコンショップ／通販部 〒321 宇都宮市竹林町503−1 TEL0286-22-9811 FAX0286-25-3970

マイコンショップ BASIC HOUSE お申し込み・お問い合せは ☎0286-22-9811(代)

64180CPU on X68000

Mach180™

X68000上でCP/M-80のソフト開発を可能にしました。

Z80上位コンパチの高機能チップであるHD64180(クロック10MHz)を使用したCPUボードとCP/M-80のBDOSエミュレータソフトの使用により、Human 68kのコマンドラインからCP/M-80用のソフトを起動させる事ができます。しかし、X68000がCP/Mマシンになる訳ではありません。Human68k上でCP/M-80のBDOSをエミュレートしますので従来と同じX68000の環境のもとで、Z80とHD64180の開発を行なう事ができるのです。

例えばX68000用のエディタを使用してプログラムの入力及びエディットを行ないます。それをCP/M-80用のアセンブラとリンカを使用してオブジェクトプログラムを生成しCP/M-80用のデバッカを使用して実行及びデバックを行なう………といった一連の開発過程を1台のX68000上で、しかもなんの切り替えの手間もなく行なう事ができます。またturboCP/M(2HD)のフロッピーディスクのアクセスを可能にしたことにより、Human68k標準のCOPYコマンドでの相互変換を実現しました。

主な特長

- CPUにHD64180(クロック10MHz/ノーウェイト)を採用、8ビット最高速
- メモリは64Kバイトを実装、64k CP/Mとして使用可能
- CP/M-80 BDOSエミュレータの使用によりBDOSレベルでのCP/M-80互換を実現
- Human68kのコマンドと同一ディスク上での混在使用が可能
- モード切り替え一切不要
- CP/MディスクドライバによりturboCP/M(2HD)のフロッピーの直接アクセスが可能
  (turboCP/M以外のCP/MではRS-232Cを利用したHEXファイルコンバータで対応)

商品名	Mach180™
型式	KGB-CPXB
商品構成	CPUボード BDOSエミュレータ ディスクドライバ 取り扱い説明書
価格	￥98,000(消費税別)

(注意)
- CP/M80のコマンドは含まれていません
- CP/M80のシステムは必要ありません
- Human68kの機能上サポートされていないBDOSコールが一部あります
- エスケープシーケンスはHuman68kに準拠します
- ハード上の制約によりアクセスできるディスクはturboCP/M(2HD)に限られます

株式会社計測技研

本社営業部／通販部
〒321 栃木県宇都宮市竹林町503-1

Tel.0286-22-9811
Fax.0286-25-3970

クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(年中無休)
AM10:00〜PM7:00(日曜・祭日はPM6:00まで)

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

基本セット X68000 PRO

- ●CZ-652C(本体・キーボード・マウス)……¥298,000
- ●CZ-602D(カラー専用ディスプレイ)……¥ 99,800
- ●CZ-8PC3(熱転写カラー漢字プリンタ)……¥ 65,800
- ●アフターバーナ(ゲームソフト)……¥ 9,200
- ●プリンター用紙・ブランクディスケット……¥サービス
- ■定価合計……¥472,800

クリエイト特価
電話にてお問合せください。

パソコン通信セット X68000 PRO HD

- ●CZ-662C(本体・キーボード・マウス)……¥408,000
- ●CZ-602D(カラー専用ディスプレイ)……¥ 99,800
- ●CZ-8PK7(24ピン漢字プリンタ)……¥122,000
- ●コミュニケーションPRO-68(高機能通信ソフト)……¥ 19,800
- ●MD-2400B(オムロン・モデム)……¥ 49,800
- ●プリンター用紙・ブランクディスケット……¥サービス
- ■定価合計……¥699,400

クリエイト特価
電話にてお問合せください。

格安基本セット X68000 PRO

- ●CZ-652C(本体・キーボード・マウス)……¥298,000
- ●CZ-603D(カラー専用ディスプレイ)……¥ 84,800
- ●アフターバーナ(ゲームソフト)……¥サービス
- ●ブランクディスケット(5"2HD・10枚)……¥サービス
- ■定価合計……¥382,800

クリエイト特価
電話にてお問合せください。

ミュージックワークセット X68000 EXPERT

- ●CZ-602C(本体・キーボード・マウス)……¥356,000
- ●CZ-602D(カラー専用ディスプレイ)……¥ 99,800
- ●CZ8PK8(24ピン136桁漢字プリンタ)……¥152,000
- ●CZ6BM1(MIDIボード)……¥ 26,800
- ●MT-32(MIDI音源ユニット)……¥ 69,000
- ●AN-S100(アンプスピーカー)……¥ 36,600
- ●MUSIC PRO(MIDI版)……¥ 28,800
- ●Musicstudio(MIDIマルチレコーディングソフト)……¥ 25,800
- ●ブランクディスケット……¥サービス
- ■定価合計……¥794,800

クリエイト特価
電話にてお問合せください。

※本広告に掲載の全商品の価格について消費税は含まれておりません。

グラフィックワークセット X68000 PRO HD

- ●CZ-662C(本体・キーボード・マウス)……¥408,000
- ●CZ-612D(0.31ピッチ・カラーディスプレイ)……¥119,800
- ●CZ-8NS1(カラーイメージスキャナ)……¥188,000
- ●CZ-6BN1(スキャナ用パラレルボード)……¥ 29,800
- ●CZ-6PV1(ビデオプリンタ)……¥198,000
- ●IO-730(カラーインクジェットプリンタ)……¥230,000
- ●Z'sSTAFF PRO-68K……¥ 58,000
- ●サイクロンEXPRESS……¥ 78,000
- ●CZ-6BE1A(1MB増設RAMボード)……¥ 38,000
- ●ブランクディスケット(5"2HD 10枚)……¥サービス
- ■定価合計……¥1,347,600

クリエイト特価
電話にてお問合せください。

大サービス・ゲーマーズセット X68000 PRO

- ●CZ-652C(本体・キーボード・マウス)……¥298,000
- ●CZ-603D(カラー専用ディスプレイ)……¥ 84,800
- ●XE-1PRO(ジョイスティック)……¥サービス
- ●CZ-8NT1(トラックボール)……¥ 13,800
- ●CZ-8NJ2(ジョイカード)……¥ 23,800
- ●ドラゴンスピリッツ(ゲームソフト)……¥ 8,800
- ●源平討魔伝(ゲームソフト)……¥ 7,800
- ●アフターバーナ(ゲームソフト)……¥ 9,200
- ●沙羅曼蛇(ゲームソフト)……¥ 8,800
- ●フルスロットル(ゲームソフト)……¥ 7,800
- ●サンダーフォース(ゲームソフト)……¥ 9,800
- ●ドッチボール(ゲームソフト)……¥ 7,800
- ■定価合計……¥480,400

クリエイト特価
電話にてお問合せください。

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

X68000シリーズ用 周辺機器・ソフトお買い得セール

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Musicstudio PRO-68K	MIDIマルチレコーディングソフト	¥ 25,800
CZ-6BE4	4MB増設RAMボード	¥138,000	NEW Print Shop PRO-68K	ポップアートツール	¥ 19,800
CZ-6BU1	ユニバーサルI/Oボード	¥ 39,800	Communication PRO-68K	高機能通信ソフト	¥ 19,800
CZ-6BG1	GP-IBボード	¥ 59,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	AI-68K	AI開発ツール	¥188,000
CZ-8NT1	トラックボール	¥ 13,800	BUSINESS PRO-68K	統合型計算ソフト	¥ 68,000
CZ-6BM1	MIDIボード	¥ 26,800	DATA PRO-68K	コマンド型リレーショナルデータベース	¥ 58,000
CZ-6EB1	拡張I/Oボックス(4スロット)	¥ 88,000	CARD PRO-68K	カード型リレーショナルデータベース	¥ 29,800
CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	TOP財務会計	プロフェッショナル財務会計ソフトウェア	¥200,000
CZ-603D	ドットピッチ0.31mm14型高解像度	¥ 84,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6TU	パソコンチューナ	¥ 33,100	Human 68K Ver2.0	開発ツールセット	¥ 9,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。●超特価販売中!

●横浜店
駅から歩いて2分!
相鉄ビル
三越
三和銀行
高島屋
ソフトクリエイト
横浜駅
至桜木町
至東京

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ
ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店(普)No.5000340

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

安心と信頼の誌上ショッピング メディアショップ お申込みは今すぐ電話かハガキで!!

株式会社 メディアショップ ハイランド　〒239 神奈川県横須賀市ハイランド3-9-6

電話でのお申込みは

お申し込みはフリーダイヤルで(料金無料)

0120-483290

お問合せは専用ダイヤルで

☎0468-483290

年中無休AM10時～PM10時

ハガキでのお申込みは

〒239
神奈川県横須賀市ハイランド3-9-6
(株)メディアショップハイランド
Ohc×係

申込書
● 商品名(商品番号)
● 支払回数
● お名前
● 生年月日
● ご住所、電話番号
● お勤め先
名称、住所、電話番号

通信販売のお申込み方法

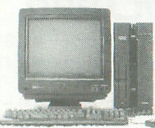
▶ 現金一括でお申込みの方

● 商品名(商品番号)及び、住所、氏名、電話番号、ご覧の雑誌名をご記入の上、代金を現金書留でお送り下さい。

● 振込をご希望の方は、必ずお振込前にお電話又はおハガキで、お知らせ下さい。
〈銀行振込〉協和銀行・久里浜支店　当座No.2945
〈郵便振替〉横浜9-42177

▶ クレジットでお申込みの方

● 電話かハガキでお申込み下さい。
クレジット申し込み用紙をお送り致しますので、ご記入の上、当社へお送り下さい。

SHARP X68000 EXPERT

●CZ-602C(FDタイプ) 標準価格 356,000円
●CZ-612C(HDタイプ) 標準価格 466,000円
●CZ-602D(ディスプレイテレビ) 標準価格 99,800円
●CZ-612D(ディスプレイテレビ) 標準価格 119,800円
●CZ-603D(ディスプレイ) 標準価格 84,800円

SHARP X68000 PRO

●CZ-652C(FDタイプ) 標準価格 298,000円
●CZ-662C(HDタイプ) 標準価格 408,000円
●CZ-602D(ディスプレイテレビ) 標準価格 99,800円
●CZ-612D(ディスプレイテレビ) 標準価格 119,800円
●CZ-603D(ディスプレイ) 標準価格 84,800円

X68000 オリジナルグッズ プレゼント!!

●X68000 スポーツタオル
●X68000 ビジネスバッグ
●X68000 ポーチ
●X68000 マウスパット

御買上げのお客様に、X68000 オリジナルグッズを1点もれなくプレゼント。

EXPERT グラフィックス

●CZ-612C(本体)……466,000円
●CZ-612D(ディスプレイテレビ)……119,800円
●CZ-8NS1(イメージスキャナー)……188,000円
●CZ-6BN1(パラレルボード)……29,800円
●AP-800(48ドットカラープリンタ)……97,800円
●#8226(インターフェイスケーブル)……8,800円
●A-400HP(ビデオデッキ)……104,800円
●CZ-221HS(NEW Print SHOP)……19,800円
●C-TRACE68(レイトレーシングソフト)……68,000円
標準価格 1,102,800円

商品番号 227	一括払価格 特別価格
初回16,500円・12,600円×47回	ボーナス60,000円×8回
初回14,600円・10,600円×59回	ボーナス50,000円×10回

EXPERT サウンド[MIDI]

●CZ-602C(本体)……356,000円
●CZ-602D(ディスプレイテレビ)……99,800円
●AN-160SP(アンプ内蔵スピーカーシステム)55,300円
●CZ-6BM1(MIDIボード)……26,800円
●MT-32(MIDI音源モジュール)……69,000円
●D-10(キーボード)……128,000円
●CZ-215MS(Sampling PRO68K)……17,800円
●CZ-247MS(MUSICPRO68K MIDI)……28,800円
標準価格 781,500円

商品番号 228	一括払価格 特別価格
初回11,000円・10,100円×47回	ボーナス50,000円×8回
初回14,100円・8,700円×59回	ボーナス40,000円×10回

EXPERT 通信・パソコンFAX

●CZ-612C(本体)……466,000円
●CZ-603D(ディスプレイ)……84,800円
●CZ-8TM2(モデムユニット)……49,800円
●VP-2000(136桁カラー漢字ドットプリンタ)……156,000円
●#8226(インターフェイスケーブル)……8,800円
●CZ-6BC1(FAXボード)……79,800円
●CZ-223CS(Communication)……19,800円
標準価格 865,000円

商品番号 219	一括払価格 特別価格
初回12,000円・9,700円×47回	ボーナス50,000円×8回
初回13,200円・8,400円×59回	ボーナス40,000円×10回

PRO ワープロ

●CZ-652C(本体)……298,000円
●CZ-603D(ディスプレイ)……84,800円
●VP-2000(136桁カラー漢字ドットプリンタ)……156,000円
●#8226(インターフェイスケーブル)……8,800円
●EW(日本語ワープロソフト)……38,000円
標準価格 585,600円

商品番号 221	一括払価格 特別価格
初回10,700円・7,300円×47回	ボーナス30,000円×8回
初回8,800円・7,000円×59回	ボーナス20,000円×10回

PRO データベース

●CZ-662C(本体)……408,000円
●CZ-612D(ディスプレイテレビ)……119,800円
●CZ-6VT1(カラーイメージユニット)……69,800円
●VP-900(80桁カラー漢字ドットプリンタ)……126,000円
●#8226(インターフェイスケーブル)……8,800円
●CZ-220BS(DATA PRO68K)……58,000円
●CZ-226BS(CARD PRO68K)……29,800円
標準価格 820,200円

商品番号 229	一括払価格 特別価格
初回12,900円・8,900円×47回	ボーナス50,000円×8回
初回10,200円・7,800円×59回	ボーナス40,000円×10回

X1, X1turbo X68000シリーズ用周辺機器

カラービデオプリンタ

●CZ-6PV1
パソコンやビデオ機器に対応、64階調(485×480ドット)で再現する、昇華性染料熱転写方式を採用。
標準価格 198,000円

商品番号 149	一括払価格 特別価格
24回	初回9,600円・7,500円×23回
36回	初回5,500円・5,200円×35回

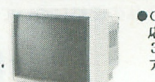
カラー イメージ スキャナー

●CZ-8NS1
高速、高精度でハイレベルな画像入力を実現 最大A4サイズの原稿をフルカラー読み取り可能
標準価格 188,000円

商品番号 188	一括払価格 特別価格
24回	初回8,100円・7,200円×23回
36回	初回7,400円・4,900円×35回

48ドット熱転写カラー漢字プリンタ

●CZ-8PC4
精緻で略字のない高品位印字。美文書もアートワークも鮮やかに、美しさの48ドットカラープリンタ。
標準価格 99,800円

商品番号 216	一括払価格 特別価格
12回	初回7,600円・7,400円×11回
24回	初回4,200円・3,900円×23回

21型カラーディスプレイ

●CU-21CD
応用分野を広げるワイド画面。3モードマルチスキャン採用アナログカラーディスプレイ。
標準価格 139,800円

商品番号 217	一括払価格 特別価格
24回	初回7,300円・5,300円×23回
36回	初回7,000円・3,600円×35回

商品名	型式	標準価格	販売価格
14型カラーディスプレイ	CZ-603D	84,800	71,900
RGBシステムチューナー	CZ-6TU	33,100	29,300
CRTフィルター	BF-68PRO	19,800	16,300
熱転写カラープリンタ	CZ-8PC3	65,800	54,000
漢字プリンタ(80桁)	CZ-8PK9	89,800	72,700
漢字プリンタ(80桁)	CZ-8PK7	122,000	98,400
漢字プリンタ(136桁)	CZ-8PK8	152,000	122,800
ハードディスク(20MB)	CZ-620H	178,000	143,700
増設用HDD(40MB)	CZ-64H	120,000	100,200
モデムユニット	CZ-8TM2	49,800	42,100

商品名	型式	標準価格	販売価格
カラーイメージユニット	CZ-6VT1	69,800	59,000
スキャナ用パラレルボード	CZ-6BN1	29,800	25,200
1MB増設RAM	CZ-6BE1	35,000	29,500
1MB増設RAM	CZ-6BE1A	38,000	32,100
2MB増設RAM	CZ-6BE2	79,800	67,300
4MB増設RAM	CZ-6BE4	138,000	116,400
ユニバーサルI/Oボード	CZ-6BU1	39,800	33,600
GP-IBボード	CZ-6BG1	59,800	50,400
増設用RS-232Cボード	CZ-6BF1	49,800	42,000
数値演算ボード	CZ-6BP1	79,800	67,300

商品名	型式	標準価格	販売価格
FAXボード	CZ-6BC1	79,800	67,300
MIDIボード	CZ-6BM1	26,800	23,200
拡張I/Oボックス	CZ-6EB1	88,000	74,200
システムラック	CZ-6SD1	44,800	37,800
スピーカーシステム	AN-160SP	55,300	48,500
カラーイメージボードII	CZ-8BV2	39,800	33,500
立体映像セット	CZ-8BR1	29,800	24,600
インテリジェントコントローラ	CZ-8NJ2	23,800	20,100
FM音源ボード	CZ-8BS1	23,800	20,100
フロッピーディスクユニット	CZ-503F	49,800	39,200

商品名	型式	標準価格	販売価格
DATA PRO68K	CZ-220BS	58,000	49,300
CARD PRO68K	CZ-226BS	29,800	25,400
Sampling PRO68K	CZ-215MS	17,800	15,300
NEW Print SHOP	CZ-221HS	19,800	16,400
Communication	CZ-223CS	19,800	16,900
C compiler	CZ-211LS	39,800	34,500
Musicstudio	CZ-237MS	25,800	22,200
MUSIC(MIDI)	CZ-247MS	28,800	24,600
OS-9/X68000	CZ-219SS	29,800	25,400
Stationery	CZ-240BS	8月発売予定	

今月の特選お買得品(限定)

SHARP X68000 ACE-HD

●CZ-611C
X68000にHDモデル登場。ますます熱くなる、パーソナルワークステーション。
●CZ-611D
15型カラーディスプレイテレビ。
標準価格 544,800円

商品番号 183	一括払価格 特別価格
48回	初回11,500円・10,500円×47回
60回	初回9,600円・8,800円×59回

SHARP X1 turbo Z III

●CZ-888C
画像取り込み、ビデオ編集ステレオFM音源。多彩な機能で広がるアートワーク。
ADVANNCED TURBO
●CZ-860D
14型カラーディスプレイテレビ。
標準価格 262,000円

商品番号 200	一括払価格 特別価格
24回	初回11,600円・9,600円×23回
36回	初回8,400円・6,600円×35回

安心と信頼 メディアショップ ハイランド

①完全保証 全国どこでもアフターケアOK
②全国無料配送 日曜配送可能
③支払回数は 予算に応じ3～36回 ボーナス併用可
④消費税 広告は全て消費税込みの価格で表示してあります
⑤FAXでも注文OK FAX：0468(48)3273
⑥その他広告以外の商品も取扱っております。お気軽にお問合せ下さい。

SHARP X68000 EXEショップ

パソコン・AV専門 O.A.ランド 大特価セール

OAランドで買わなきゃ損をする！

※4月1日より消費税を課税させていただきます。尚、表示価格は税別表示です。詳しくは、お電話下さい。

セール期間 '89 8・16→9・15

残暑なんかコワくないッ!! 大放出!!

NEW ランド特選 SHARP X68000EXPERT・EXPERT HDセット

X68000EXPERT HDセット

40MB HDD内蔵 2MB RAM

- CZ-612C ……………………定価￥466,000
- CZ-612D ……………………定価￥119,800
- MD-2HD 20枚サービス

ゲームソフト 5ゲームプレゼント

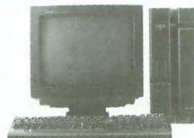
他店には負けません!! 合計定価￥585,800

現金大特価!! 安いぞ

X68000EXPERTセット

2MB RAM内蔵

- CZ-602C ……………………定価￥356,000
- CZ-612D ……………………定価￥119,800
- MD-2HD 20枚サービス

ゲームソフト 5ゲームプレゼント

OAランドで買わなきゃ損をする！ 合計定価￥475,800

現金大特価!!

大推選!!

NEW X-1ターボZⅢセット

CRTクリーナー キーボードカバープレゼント

安すぎてゴメンなさい！

Ⓐセット

- CZ-888CBK…定価￥169,800
- CZ-880DBK‥ 定価￥109,800
- CZ-6ST1-B ……定価￥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計定価￥275,400

現金価格 特価中TEL下さい

Ⓑセット

- CZ-888CBK…定価￥169,800
- CZ-830DBK…定価￥ 98,000
- CZ-6ST-1B ……定価￥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥273,600

合計価格 特価中TEL下さい

NEW SHARP X68000 PRO・PRO HDセット

ゲームソフト 5ゲームプレゼント

X68000PROセット

- CZ-652C ………定価￥298,000
- CZ-612D ………定価￥119,800
- MD-2HD 20枚サービス

合計定価￥417,800

現金特価!! TEL下さい。

X68000PRO-HDセット

- CZ-662C ………定価￥408,000
- CZ-612D ………定価￥119,800
- MD-2HD 20枚サービス

合計定価￥527,800

現金特価!! TEL下さい。

X68000 お買徳!! X-1TWIN

展示新同品

- CZ-611C(GY)
- CZ-611D(GY)

2セット限り

現金特価 ￥328,000

新同品

- CZ-830C 定価￥99,800

PCエンジン内蔵

現金特価￥38,000

周辺機器コーナー

X1用

- CZ-8BV2…定価￥ 39,800▶特価￥ 31,000
- CZ-8BR1…定価￥ 29,800▶特価￥ 23,000
- CZ-8DT2…定価￥ 44,800▶特価￥ 35,000
- CZ-8BS1…定価￥ 23,800▶TEL下さい
- CZ-8TM2…定価￥ 49,800▶特価￥ 38,000
- CZ-8EB3…定価￥ 33,800▶特価￥ 27,000

X68000用

- CZ-6PU1A‥定価￥ 38,000▶特価￥ 30,000
- CZ-6BM1…定価￥ 26,800▶特価￥ 21,000
- CZ-6BE1…定価￥ 88,000▶特価￥ 69,800
- CZ-6VT1…定価￥ 69,800▶TEL下さい
- CZ-8NS1…定価￥188,000▶特価￥149,000
- CZ-6BC1…定価￥ 79,800▶特価￥ 63,000

プリンターセットコーナー

1. CZ-6PU1(カラービデオプリンター)定価￥198,000▶特価￥152,000
2. CZ-8PC3(カラープリンター)……定価￥ 65,800▶特価￥ 53,000
3. CZ-8PK8(ドットプリンター)……定価￥152,000▶特価￥115,000
4. CZ-8PK7(ドットプリンター)……定価￥122,000▶特価￥ 93,000
5. PC-PR201TH(カラープリンター)·定価￥145,000▶特価￥103,000
6. PC-PR201G(ドットプリンター)…定価￥158,000▶特価￥ 99,000

その他、周返機器・プリンター ソフトウェアー 20%～25% OFF!!

X68000用ソフトウェアー・コーナー

1. CZ-212BS(BUSINESS)……定価￥ 68,000▶特価￥ 53,000
2. CZ-220BS(DATA)……定価￥ 58,000▶特価￥ 45,000
3. CZ-215MS(Sampling)……定価￥ 17,800▶特価￥ 13,800
4. CZ-221HS(NEW Print Shop)‥定価￥ 10,800▶特価￥ 15.500
5. CZ-227BS(TOP財務会計)……定価￥200,000▶特価￥158,000
6. CZ-226BS(CARD)……定価￥229,800▶特価￥ 23,000
7. CZ-223CS(Communication)……定価￥ 19,800▶特価￥115,500
8. CZ-213MS(MUSIC)……定価￥ 18,800▶特価￥ 14,800
9. CZ-211LS(C compiler)……定価￥ 39,800▶特価￥ 31,000
10. C-TRACE(キャスト)……定価￥ 68,000▶特価￥ 52,000
11. EW(イースト)……定価￥ 38,000▶特価￥ 29,000

■ハードディスク ■特価品もありますのでTEL下さい。

- アイテック IT-MJ4(I/F付)……特価￥98,000
- アイテック IT-MJ4 C(I/F付)……特価￥109,000
- ウィンテック HD-404HS(I/F付)……特価￥108,000
- コンピュータ CRC-MH4(I/F付)……特価￥70,000
- スナイパー SR-340II(I/F付)……特価￥78,000
- アイテック ITH-320S(I/F付)……特価￥79,800
- ウィンテック HD-202(I/F付)……特価￥58,000
- スナイパー SR-520(I/F付)……特価￥55,000
- コンピュータ CRC-HD2A(I/F付)……特価￥62,000
- ロジテック LHD-32NR(I/F付)……特価￥80,000

今月の特価品 各一台限り

その他、いろいろありますのでTEL下さい!!

■A級品(美品・POP品) ■B級品(キズ少々) ■C級品(キズ有り)

		A級品	B級品	C級品
X68000シリーズ	CZ-611C	￥250,000より	￥245,000	￥238,000
	CZ-652C	￥219,000より	￥212,000	￥203,000
	CZ-611D	￥ 90,000	￥ 86,000	￥ 80,000
	CZ-603	￥ 58,000	￥ 55,000	￥ ——
X-1シリーズ	CZ-888C	￥ 99,800より	￥ 90,000	——
	CZ-822C	￥ 24,000より	￥ 20,000	——
	CZ-880D	￥ 75,000	￥ 71,000	——
	CZ-830C	￥ 37,000	￥ 33,000	——
X-1プリンター	CZ-8PC3	￥ 48,000	￥ 45,000	￥ 42,000
	CZ-7PK7	￥ 83,000	——	——
	CZ-8PK8	￥109,000	￥105,000	——
	CZ-6PV1	￥138,000	￥134,000	￥125,000

その他、いろいろありますので、TELください。

中古パソコン(価格・在庫は変動します。予約は5日以内といたします。)

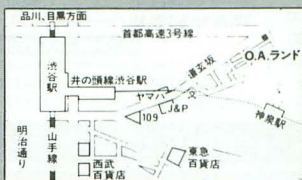
- PC-9801VX2ト…… ￥220,000より
- PC-9801VX2…… ￥195,000より
- PC-9801VM2…… ￥158,000より
- PC-9801VF2…… ￥ 98,000より
- PC-9801M2…… ￥138,000より
- PC-9801F2…… ￥ 78,000より
- PC-9801UV21…… ￥138,000より
- PC-98LTM1(640KB)‥ ￥ 89,000より
- PC-286モデル0…… ￥168,000より
- PC-286V-STD…… ￥202,000より
- X-68000…… ￥188,000より
- PC-8801mkII30…… ￥ 35,000より
- PC-8801mkIISR…… ￥ 73,000より
- PC-8801mkIIFR30‥ ￥ 68,000より
- PC-8801mkIIMR…… ￥ 88,000より
- PC-88VA…… ￥148,000より
- PC-8801mkIIFH30‥ ￥ 85,000より
- PC-8801FA…… ￥108,000より
- X-1Gモデル30…… ￥ 25,000より
- X-1ターボII…… ￥ 68,000より
- FM-77D2…… ￥ 28,000より
- FM-77AV2…… ￥ 42,000より
- FM-77AV20…… ￥ 52,000より

通信販売のご案内

全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。

〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1～60回払で月々5,000円よりご自由に設定できます。

- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…毎日午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

㈱オーエー・ランド

〒150 東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855

FAX (03)770-7080

関東エリアの送料は、1個につき￥1,000です。

クレジット
金利大幅ダウン!!

JDMA 安心と信頼のシステムで新時代を切り開く

X68000

EXPARTシリーズ
・PROシリーズ 新登場!!

・オリジナルOS「Human68k ver. 2.0」を搭載
・40MBハードディスクドライブを内蔵
・メインメモリ2MB標準装備(EXPERTシリーズ)
・拡張I/Oスロット4スロット内蔵(PROシリーズ)

☆注文No.A-0921
SHARP CZ-602C ￥356,000
SHARP CZ-602D ￥99,800
標準価格合計 ￥455,800
現金特別価格 ~~￥455,800~~
大特価にて提供中

☆注文No.A-0922
SHARP CZ-612C ￥466,000
SHARP CZ-602D ￥99,800
標準価格合計 ￥565,800
現金特別価格 ~~￥565,800~~
大特価にて提供中

☆注文No.A-0923
SHARP CZ-652C ￥298,000
SHARP CZ-602D ￥99,800
標準価格合計 ￥396,800
現金特別価格 ~~￥396,800~~
大特価にて提供中

☆注文No.A-0924
SHARP CZ-662C ￥408,000
SHARP CZ-602D ￥99,800
標準価格合計 ￥507,800
現金特別価格 ~~￥507,800~~
大特価にて提供中

当社はX68000 PRO SHOPです。

■周辺機器　大特価にて提供中

品番	品名・内容	定価
CZ-602D	15型カラーディスプレイテレビ	￥99,800
CZ-612D	15型カラーディスプレイテレビ	￥119,800
CZ-603D	14型カラーディスプレイ	￥84,800
CZ-6ST1	601D・611D用チルトスタンド	￥5,800
CU-21CD	21型カラーディスプレイ	￥139,800
CZ-6TU	RGBシステムチューナー	￥33,100
BF-68PRO	601・611・603用CRTフィルター	￥19,800
CZ-6VT1	カラーイメージユニット	￥69,800

型番	品名・内容	定価
CZ-8NS1	カラーイメージスキャナ	￥188,000
CZ-6BN1	スキャナ用パラレルボード	￥29,800
CZ-6BE1A	1MB増設RAMボード(内蔵用)	￥38,000
CZ-6BE2	2MB増設RAMボード(内蔵用)	￥79,800
CZ-6BE4	4MB増設RAMボード(内蔵用)	￥138,000
CZ-6BU1	ユバーサルI/Oボード	￥39,800
CZ-6BG1	GP-IBボード	￥59,800
CZ-6BF1	増設用RS-232Cボード(2ch)	￥49,800

型番	品名・内容	定価
CZ-6BP1	数値演算プロセッサボード	￥79,800
CZ-6BC1	FAXボード	￥79,800
CZ-6BM1	MIDIボード	￥26,800
CZ-6EB1	拡張I/Oボックス(4スロット)	￥88,000
CZ-6PV1	カラービデオプリンタ	￥198,000
CZ-6BU1	ユバーサルI/Oボード	￥39,800
CZ-620H	ハードディスクユニット(20MB)	￥178,000
AN-S100	アンプ内蔵スピーカーシステム(2本1組)	￥36,800

■ソフトウェア　大特価にて提供中

メーカー名	型番	品名・内容	定価
SHARP	CZ-212BS	BUSINESS PRO-68K	￥68,000
SHARP	CZ-220BS	DATA PRO-68K	￥58,000
SHARP	CZ-226BS	CARD PRO-68K	￥29,800
SHARP	CZ-214MS	SOUND PRO-68K	￥15,800
SHARP	CZ-213MS	MUSIC PRO-68K	￥18,800
SHARP	CZ-215MS	Sampling PRO-68K	￥17,800

メーカー名	型番	品名・内容	定価
SHARP	CZ-237MS	Musicstudio PRO-68K	￥25,800
SHARP	CZ-247MS	MUSIC PRO-68K〔MIDI〕	￥28,800
SHARP	CZ-221HS	NEW Print Shop PRO-68K	￥19,800
SHARP	CZ-223CS	Communication PRO-68K	￥19,800
SHARP	CZ-211LS	C compiler PRO-68K	￥39,800
SHARP	CZ-219SS	OS-9/6800	￥29,800

メーカー名	型名	品名・内容	定価
イースト	EW	日本語ワープロ	￥38,000
アンスコンサルタンツ	彩CRONE68K	グラフィックツール	￥58,000
CAST	C-TRACE68	グラフィックツール	￥68,000
ツァイト	Z'sSTAFF PRO	グラフィックツール	￥58,000
電波新聞社		ドラゴンスピリット	￥8,800
テクノソフト		サンダーフォースII	￥9,800

●どこよりもお得な高額下取り実施中!!　セットの組合わせは自由自在、ぜひご相談下さい。

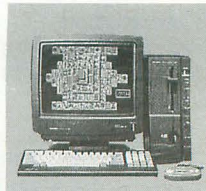
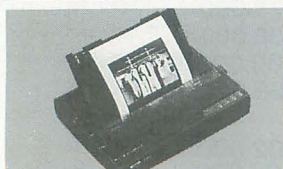
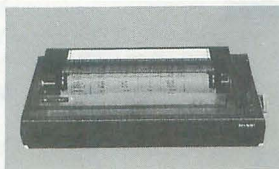
X1 turbo Z III

画像取り込み、ビデオ編集、ステレオFM
音源、多才な機能でひろがるアートワーク。

☆注文No.A-0925
SHARP CZ-888C-BK ￥169,800
SHARP CZ-860D-BK ￥92,200
標準価格合計 ￥262,000
現金特別価格 ~~￥262,000~~
大特価にて提供中

X1 twin

HEシステム(PC Engine)
搭載で楽しさ2倍

☆注文No.A-0926
SHARP CZ-830C-BK ￥99,800
SHARP CZ-830D-BK ￥90,600
標準価格合計 ￥190,400
現金特別価格 ~~￥190,400~~
大特価にて提供中

●どこよりもお得な高額下取り実施中!!　セットの組合わせは自由自在、ぜひご相談下さい。

☆注文No.B-0923
SHARP CZ-8PC3 ￥65,800
現金特別価格 ~~￥65,800~~
大特価にて提供中
■お支払例
①￥10,000×6回〔ボーナス〕無し
②￥3,200×20回〔ボーナス〕無し

☆注文No.B-0924
SHARP CZ-8PK6 ￥159,000
現金特別価格 ￥59,800
■お支払例
①￥6,500×10回〔ボーナス〕無し
②￥3,400×24回〔ボーナス〕無し

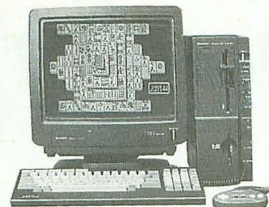
☆注文No.B-0925
SHARP CZ-8PC4 ￥99,800
現金特別価格 ~~￥99,800~~
大特価にて提供中
■お支払例
①￥9,500×10回〔ボーナス〕無し
②￥3,000×36回〔ボーナス〕無し

☆注文No.B-0932
"インテリジェントコントローラ"
SHARP CZ-8NJ2 ￥23,800
現金特別価格 ~~￥23,800~~
大特価にて提供中

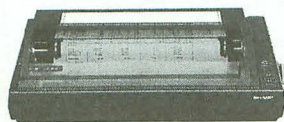
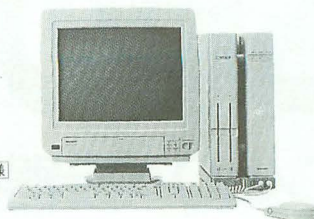
●どんな問い合わせにも親切に対応いたします。

全商品保証付 中古も6ヶ月の保証期間だから安心です。
クレジットでOK カレッジクレジットも取扱います。
全国無料配送 お買上1万円以上、配達料はいただきません。
日曜配達可 留守の多い方でも安心です。
ショールーム Xシリーズ展示中。
高額買取り 電話1本で即、現金お支払い。
代金引換えシステム 商品到着時の代金支払いでOK。
ボーナス一括払い 商品は即お手元へ、お支払いはボーナス時に。

話題の新製品が全国どこでも電話で買える!!
03(797)1221

中古在庫リスト

SHARP
CZ-830C
(X-1Twin)
¥99,800➡¥46,000

SHARP
CZ-822C
(X-1Gモデル30本体) 新品同様
¥118,000➡¥29,800
X-1Gモデル30RFコンバータセット
(本体+AN-58C) 新品同様
¥120,980➡¥32,600

SHARP
CZ-611CGY
(X68000ACEHD) 新品同様
¥399,800➡¥278,000
X68000 ACEHD
ディスプレセット
(本体+CZ-611DGY) 新品同様
¥533,800➡¥370,800

SHARP
CZ-8PK6 新品同様
(15インチ漢字プリンタ)
¥159,000➡¥59,800

SHARP
CU-14GD 新品同様
(PC用14インチ400Lアナログディスプレイ)
¥69,800➡¥54,800

SHARP
CU-14CD
(14"カラー4050文字) 新品同様
¥84,800➡¥52,800

SHARP

本体

品名	定価	価格
CZ-812C(X-1F model20)	¥139,800	¥26,000
CZ-822C(X-1F model30)	¥118,000	¥28,000
CZ-822CB(X-1G model30) 新品同様	¥118,000	¥29,800
CZ-601C(X68000ACE) 新品同様	¥319,800	¥238,000
CZ-611C(X68000ACEHD) 新品同様	¥399,800	¥278,000
MZ-1500	¥89,800	¥15,000
MZ2521	¥198,000	¥45,000

ディスプレイ

品名	定価	価格
12M-18B(12"グリーン4050文字)	¥44,800	¥20,000
14M-142C(14"カラー2000文字)	¥99,800	¥22,000
CU-14BD(14"カラー2000/4000文字)	¥64,800	¥42,000
CU-14GD(14"カラー4050文字) 新品同様	¥69,800	¥54,800
CU-14FD(14"カラー4050文字) 新品同様	¥74,800	¥59,800
CU-14CD(14"カラー4050文字) 新品同様	¥84,800	¥52,800
CZ-880DBK(14"カラー2000/4000文字TV) 新品同様	¥102,000	¥75,000
CZ-603D(15"カラーX68000) 新品	¥84,800	¥68,000
CZ-611D(15"カラー3モードスキャンディスプレイTV) 新品同様	¥134,000	¥92,800
各種4050文字ディスプレイ		¥42,000～

ディスクドライブ・プリンタ・他

品名	定価	価格
CZ-52F(X1用内臓ドライブ)	¥34,800	¥9,800
CZ-503F(X1用増設1ドライブ)	¥49,800	¥25,000
CZ-81P(CZ-801C専用カラープロッタプリンタ)	¥54,800	¥10,000
CZ-8PK4(10"24ドット漢字プリンタ)	¥158,000	¥45,000
CZ-8PK6(15"24ドット漢字プリンタ) 新品	¥159,000	¥59,800
CZ-8PK7(10"24ドット漢字プリンタ)	¥122,000	¥52,000
MZ-1P07(10"ドットプリンタ)	¥79,800	¥22,000
MZ-1P17(10"24ドットカラー漢字サーマルプリンタ)	¥79,800	¥30,000
CZ-8SS2(システムスタンド) 新品	¥5,500	¥3,500
CZ-8BS1(FM音源ボード) 新品	¥23,800	¥20,000

その他各種在庫をとりそろえております。御気軽にお問い合わせ下さい。

6つの安心のアフターサービス

1 C.B.クラブ

■あなたも今すぐ会員に!!

当社で商品をお買い上げの方全員に、C.B.クラブカードを無料でお送り致します。このカードをお持ちの方なら次の買い換え時や、付属品の購入時に会員特別価格でご購入になれます。

2 C.B.サポートホットライン

☎03(797)1234

■トラブルへの対応!!

当社でコンピュータをお買い上げいただいたお客様に万一、トラブルが発生した場合、このホットラインで親切に対応いたします。

3 C.B.レスキューシステム

■迅速なサポート体制!!

お客様のお手元でトラブルが発生した場合、当社より引取りにお伺い致します。万一、お買いになった機械が故障しても安心です。

4 C.B.クイック・チェンジシステム

■新品交換体制も万全!!

お買い上げになったパソコンが、万一初期不良でも安心です。商品到着後7日以内にご連絡いただければ、新品と交換致します。

5 RX2アフターサポート

■PC-9801愛好家にお得です!!

NEC RX2をお買い上げいただいたお客様に保証期間中、万一故障があった場合無料で代品を貸出します。

6 C.B.Q&Aホットライン

☎03(797)1233

■素朴な疑問何でもどうぞ!!

ハードウェア、ソフトウェアに関するご質問なら内容を問わずどなたからでも親切に、ご相談をお受け致しております。

●コンピュータを売りたい方、査定をご希望の方、その他買取りに関するご相談は●

買取り専用デスク 03(797)1231

- ●電話一本で高額下取り、即商品はお手元へ!
- ●あなたの不要になったパソコンを電話一本で査定し買取ります。
- ●掲載の商品以外も取り扱っております。
- ●ビジネスソフトスクール受講者受付中!
  お気軽にお電話下さい。

▼本社注文デスク
03(797)1221

コンピュータバンク

株式会社パシフィックコンピュータバンク 〒150 東京都渋谷区渋谷1-6-8 井上ビル 営業時間/平日AM9:30～PM9:00 土・休日AM9:30～PM8:00 年中無休

●クレジット価格に消費税は含まれておりますが、現金特別価格には含まれておりません。別途消費税がかかります。

Let's CALL
超優良中古パソコンが電話一本で買える!!
03(797)1221

変えたJ&P。

Joshin Computer Store
J&P

名古屋 大須 **大須店**

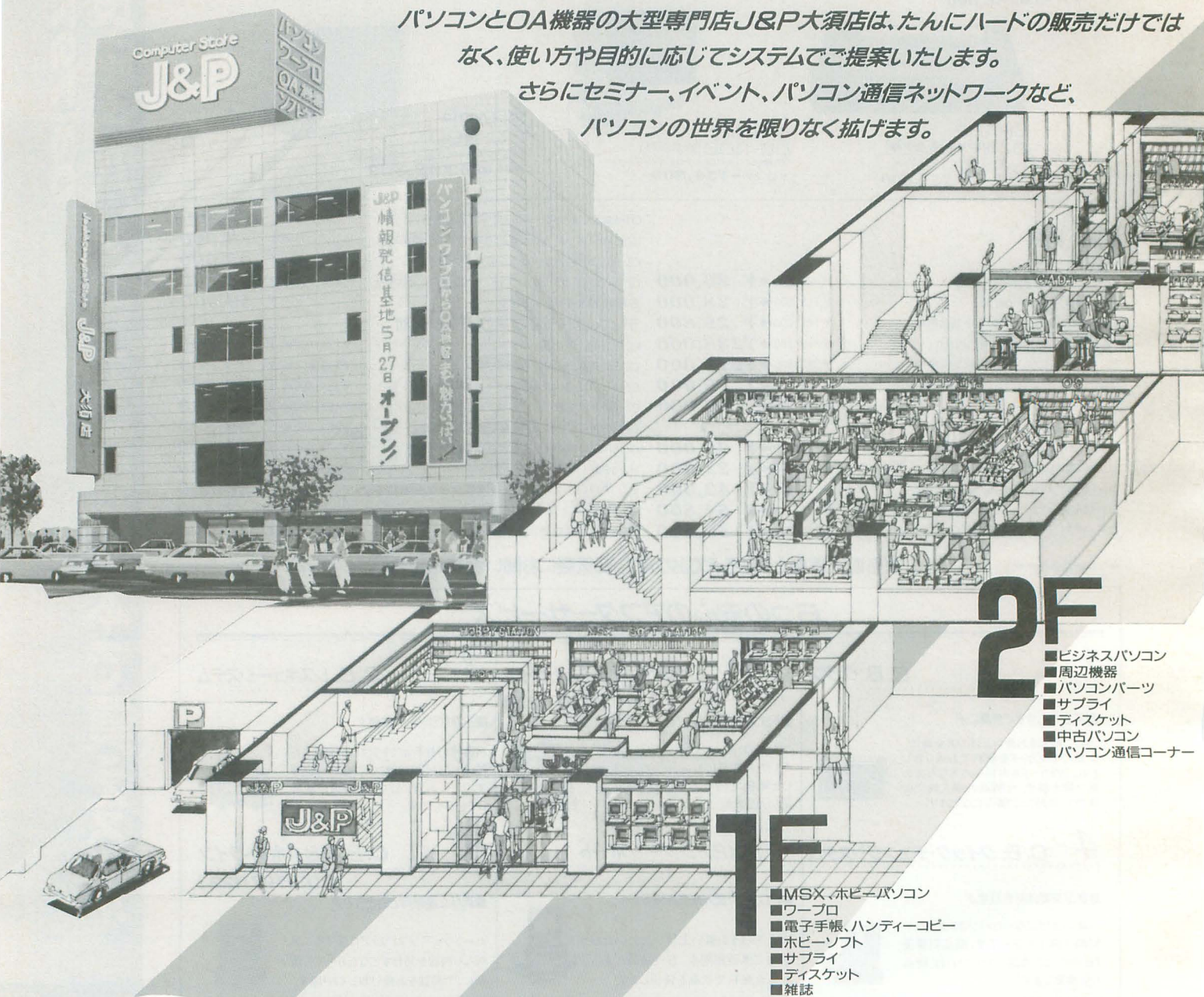
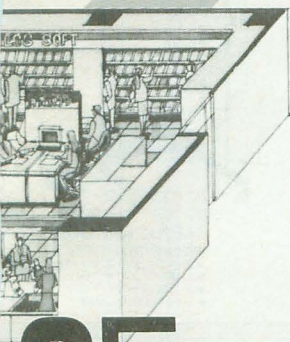
〒460 名古屋市中区大須4丁目2-48
☎(052)262-1141

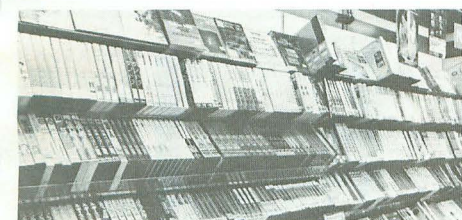
P 立体駐車場完備

パソコン教室

パソコン入門からビジネスコースまでの幅広いカリキュラムを用意しています。

CADコーナー

あらゆるプロユースにお応えできる本格的CADコーナー。ハードからソフト、周辺機器まで、目的・ご予算に応じてシステムで提案いたします。

3F

- 業種、業務別システム
- ビジネスソフト
- CADコーナー
- 消費税コーナー
- ラップトップパソコン・アップル
- FAX・PPC
- 伝票・帳票用紙
- 専門書籍
- パソコン教室

ハードウェア

MSXからIBMまで人気のハードウェアをはじめ、プリンタ、ハードディスクなどの周辺機器も勢揃い。さまざまなニーズに的確にお応えします。

ソフトウェア

人気のホビーソフトをはじめ、日本語ワープロ、販売管理、顧客管理、データベース、各種OSなどあらゆる用途の国内外のソフトウェアが勢揃い。

★その他、定期的にOAフェアや各機種ユーザー会など開催いたしております。

パソコン通信ネットワークサービス
J&P HOT LINE

「証券情報」「ソフト新作情報」「新刊書籍」「ワープロ文例集」「パソコン新製品情報」をはじめ、楽しい情報が居ながらにして得られます。

J&P HOT LINE概要

(全国90ヶ所のアクセスポイントにより、日本全国どこからでも同一料金でご利用いただけます。)

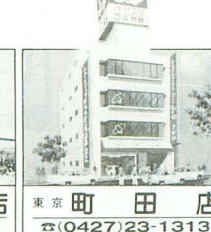
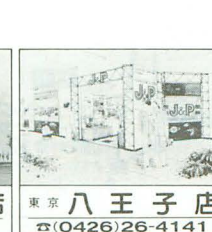
- 入会金/3,000円(スタータキットの代金で充当されます)
- 接続料/3分あたり20円(アクセスポイントまでの電話代は含みません)
- 営業時間/毎日AM6:00～翌AM2:00(1日20時間)
- J&P HOT LINE会員数/約21,000人(1989年6月末現在)
- お申し込み方法/スタータキットをご購入いただき、申込書をご返送いただいた時点で入会とさせていただきます。(アクセスはスタータキット購入日から可能です)

〈お問い合せ先〉上新電機㈱ J&P HOT LINE事務局
〒556 大阪市浪速区日本橋5-6-7 ☎(06)632-2521

会費無料 **J&P会員募集!**

パソコン・ワープロをお持ちの方ならどなたでもご入会いただけます。

■入会特典:パソコンソフトの特別割引・パソコンパーツの特別割引・パソコン教室受講料の特別割引・新製品情報の提供・セミナー、セールへのご招待など特典がいっぱいです。

入会ご希望の方は、店頭までお越しください。

パソコン通信
J&P HOT LINE でもお申し込みいただけます。

J&Pメールショッ

■600㎜幅パソコンラック

X9-1
パソコンラック
シグマPW-833 ¥29,500
J&P特価 15,800円
W600×D640×H1280㎜

¥32,800を J&P特価 15,800円
X9-2
パソコンラック
エレコムDS-20
W650×D700×H1260㎜
●ロック式キャスター付
●コード落しボックス付

¥43,000を J&P特価 19,800円
X9-3
パソコンラック
エレコムPD-02
W640×D700×H1305㎜
●ロック式キャスター付
●コード落しボックス付
●中棚が2枚でキーボード収納棚又はペーパー置きにできます。

X9-4
パソコンラックER-600
オーバートップデスクER-606付
エレコムER-600
オプションプリンタ台ER-606
¥合計38,000
J&P特価 15,800円
W650×D625×H1355㎜

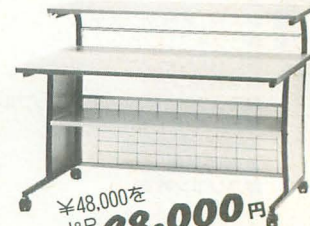
X9-5
パソコンラック
サンワSR-106 ¥36,800
J&P特価 16,800円
W620×D700×H1265㎜
●ロック式キャスター付●コード落しボックス付●コンセント付
●キャスター付
●2Pコンセント2個付

■900㎜幅パソコンラック

パソコンデスク X9-6
エレコムPD-99
J&P特価 29,800円 ¥48,000
W900×D700×H1280㎜
●ロック式キャスター付●コード落しボックス付●2Pコンセント2個付●B4判引き出し別売¥3,500

¥49,800を J&P特価 42,900円
X9-7
ワーク・ステーション・デスク
シグマPW-9300
W900×D740×H875
●コンセント、手許スイッチ付●5"フロッピィサイズ引出し●データスタンド、ペーパーストレイジ、OAチェアー別売

■1,200㎜幅パソコンラック（ゆったり派）

パソコンデスク
エレコムPD-120 ¥48,000 X9-8
J&P特価 31,500円
W1200×D700×H820～1180㎜
●ロック式キャスター付●オーバートップ調節可●2Pコンセント2個付●B4判引き出し別売 ¥3,500

¥48,000を J&P特価 28,000円
パソコンデスク X9-9
エレコムER-1200
W1200×D700×H820～1180㎜
●ロック式キャスター付
●オーバートップデスク高さ調節可

■OAチェアー

X9-10
¥12,000を J&P特価 5,800円
OAチェアー
スターL-395
●張地布、ネジ式座面上下調節●キャスター付●色/ブルー、ブラウン、グレー

レバーひとつでらくらく高さ調整！

X9-11
OAチェアー
エレコムCCF-30 ¥32,000
J&P特価 18,800円
●張地/布、ガス式座面上下調節
●キャスター付

■オプション

データスタンド
エレコムSO-550
¥7,000
J&P特価 4,000円
データホルダー
●A3、B4ヨコ型タイプ
●カーソル・クリップ付
X9-16

マウステーブル X9-14
エレコムMT-1、MT-2
J&P特価 3,500円 ¥5,500
MT-1/対応機種PD-01、02
MT-2/対応機種DS-10、20、ER-600、900、PD-99

X9-15
モニタースタンド
M.S.C. YU-M11 ¥29,800
J&P特価 19,800円
耐久重量60kg
14、15インチモニター用
机の上が広々と使えます。

X9-17
CRTフィルター HOYA
●アイテックフィルターF2B ¥15,000
14インチモニター用 J&P特価 10,000円
●スタンダードタイプ
●アイテックフィルターA(エース) ¥19,000
14インチモニター用 J&P特価 15,200円
静電気防止タイプ
●アイテックフィルターU ¥28,000
14インチモニター用 J&P特価 23,000円
電磁波防止タイプ

■その他のラック

ワープロユーザーにおすすめ！

X9-12
マルチラック
エレコムERX-7 ¥15,000
J&P特価 9,800円
W600×D700×H650㎜

ラップトップパソコンユーザーにおすすめ！

都会派ラック

X9-13
パソコンラック
エレコムPD-500 ¥15,000
J&P特価 12,800円
W500×D625×H835㎜
ラップトップパソコンにピッタリ！間口500㎜サイズの省スペースラック

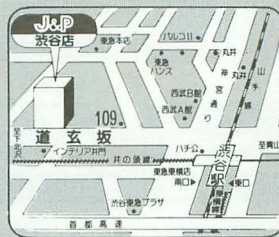
¥はメーカー標準価格です。

全国無料配達

ピング

メールショッピングのお申し込みは J&P 渋谷店で承ります。

フロアーごあんない

4F	パソコン教室 ●パソコン入門コース●BASIC上級コース ●BASIC初級コース●各種ビジネスコース
3F	OA機器 ●ビジネスパソコン●ワードプロセッサ ●ビジネスソフト●OAサプライ ●ハンドヘルトコンピュータ
2F	ビジネスパソコン ●パソコン●ディスプレイ ●プリンタ●専門書籍 ●パソコンアクセサリー
1F	ホビーのパソコン ●ホビーパソコン●MSX ●ゲームソフト●学習ソフト

Personal Computer Store
J&P 渋谷店
東京都渋谷区道玄坂2丁目28番4号(〒150)
☎(03)496-4141〈水曜定休〉

●電子手帳

X9-18

■シャープ電子手帳PA-8500
メーカー 標準価格¥28,000 **¥24,800**
約660人分のデータ管理にカレンダー・スケジュール・メモ・計算・時計機能搭載。さらにオプションカードで面白さが広がります。

〈PA-8500を面白くするICカード〉

X9-19

●シャープPA-7C40
英和辞書カード [4行表示] **¥16,000**
本格派の辞書に匹敵する4万語を収録。

●シャープPA-7C41
国語辞典カード [4行表示] **¥16,000**
読みを入力すれば漢字の変換、意味、熟語、用語、類語、対語を表示。

●シャープPA-7C43
珠玉格言集カード [4行表示]
メーカー標準価格¥10,000
¥9,000
約1700句の名言・格言を収録。中国古典、日本古典より精選。

●シャープPA-7C10
電話帳/住所録カード [4行表示]
メーカー標準価格¥10,000
¥9,000
電話帳なら約330人、住所録なら約170人分を記憶。メモリー・バックアップ付。

●シャープPA-7C3 メーカー標準価格¥7,000
電訳機、6ヶ国語カード **¥6,300**
日本語、中国語、韓国語、英語、フランス語、スペイン語の会話文約350例と単語約610語を収録

X9-20

■ハル研究所 HAL-CATCH
¥11,000 パソコンで電子手帳のデータ管理を実現。大きな画面で整理したデータを再度電子手帳へ転送可能。
●電子手帳は別売

●パソコン通信

J&P HOTLINE入会キットがセットなので買ったその日からアクセス可能

X9-21

■オムロンモデム MD-12FS
＋J&P HOTLINEスタータキット
¥22,800
AC・DC両用タイプなのでラップトップパソコンに最適。1200・360bps全二重。

X9-22

高速通信セット

■アイワモデム
PV-A24MNP5
＋J&P HOTLINEスタータキット
¥49,800
2400・1200・300bps全二重。高速通信に対応した人気モデムにJ&P HOTLINE入会キットをセット。

●ハンディーコピー

X9-23

■ゼロックスハンディコピー
写楽アルファII
¥56,800
104mm幅、拡大縮小機能のついた実務機。

●パソコンアクセサリー

X9-24

■5インチ2HD
ディスケット
スリーエム MD2D256EX
[30枚で] **¥4,200**

X9-25

■3.5インチ2HD
ディスケット
データライフ
MF-2HDPRO
[20枚で] **¥8,500**

X9-26

■フロッピィケース
5インチ用
エレコム FP5-180
メーカー標準価格¥5,800
¥5,000

X9-27

■OAライト
MEIKO EP-303
メーカー標準価格12,800
¥6,980
パソコンライトU
遮光板付

■ジョイカード X9-28
スピタル JOYCONT-turbo-3
メーカー標準価格¥2,200
¥2,000 X-6800用 ダブルトリガーで即連射！

X9-29

■フロッピィケース3.5インチ用
エレコム FP35-180 メーカー標準価格¥5,800
¥5,000

■キーボード
防塵カバー X9-30
エレコム X-6800エキスパート
プロ用があります。
¥2,800
シリコン製。PKB-X68

お申し込み方法

右の注文書にご希望商品の注文No.および必要事項ご記入の上、現金書留にて J&P 渋谷店までお申し込みください。現金受領後、発送いたします。
また、J&P HOTLINE会員の方は、ショッピングコーナーでもお申し込みいただけます。

●価格には消費税が含まれております。
●記載商品以外のご注文も承ります。
詳しくはお電話にてお問い合わせ下さい。
☎(03)496―4141 定休：毎週水曜日

キリトリ線

現金書留申込み用紙

おところ 〒□□□□□	注文No.	数量	金額
	X9-　（　　）		円
	X9-　（　　）		円
TEL　（　　　）	合計		円
おなまえ　　　様	お手持ちのパソコン		

お申込み先：東京都渋谷区道玄坂2丁目28番4号(〒150) J&P 渋谷店メールショッピング係

他と比べてみて下さい。アフターケアの質が違います。

業界初、IPL独自の3倍保証
例えばメーカー保証12ヶ月の商品なら36ヶ月。

初期不良交換期間1ヶ月
集配便を伺わせ、新品と交換させて頂きます。

最長84回のクレジット
お支払い回数は、1回払いから最長84回まで可能。

全国無料配送
配送日の指定。さらに特急配送（有料）も可能。

ステップアップクレジット
初回は低額、ライフプランに合せてステップアップ。

カレッジクレジット
学生・専門学校や新社会人の方も即クレジット。

ゆとりの8ヶ月先払い
お支払いは翌月一括払いから8ヶ月先まで自由に設定。

冬のボーナス一括払い
今から冬の一括払いもOK。夏冬2回払いも可能。

キーボードレッスンを進呈
PC-98、エプソン286シリーズに無料添付。

添削付通信教育講座（無料）
安心してお使い頂けるよう、全システムに無料添付。

ビギナーズホットライン
初心者の方のために、無料相談窓口を常設。

テクニカルホットライン
購入後のハード及びソフトのサポート窓口。

初心者の方でも安心！ 通信教育講座IPL独自のサポート体制

70,000人＋αの結論。

IPLだから、1ヶ月間の初期不良交換サービス！

「少しでも安く」と、いろいろ探してやっと見つけたパソコンが、いざ動かしてみると動かない！ でも購入してから一週間以上。結局、高くついてしまった。よくある話しですが、IPLなら1ヶ月間の初期不良交換期間を設けてあります。欲しい物、好きな物が安心して購入出来るのが特徴です。

WIDE SUPPORT

メーカー保証＋IPL保証×2＝3倍保証

メーカー保証12ヶ月の商品なら36ヶ月の保証とグッと長期間の保証を実施。末長く安心していただけるよう、IPLが成し得たワイドなサポート体制。
（もし実費で修理したらこんなに費用がかかります：プリンタヘッド交換¥29,500以上/98シリーズメインボード交換¥21,600以上/ドライブ交換¥13,200以上）

●IPLだからこそ初期不良への保証も万全。交換期間も1ヶ月ともっとも長期間です。

●安心のサポート。IPLキーボードレッスン無料で添付。
目でさがさず、指がキーボードを確実に覚えて、プログラミング上達に格段の差がつくレッスン用ソフト（¥9,800）をNEC PC-98、EPSON286シリーズ（但し、ラップトップを除く）に無料で添付。

●IPLの実績から実戦・初の通信教育制度。
初めてコンピュータを手にしたその日から安心してお使いいただける様、IPL独自の添削付通信教育制度です。もちろん受講料は無料です。

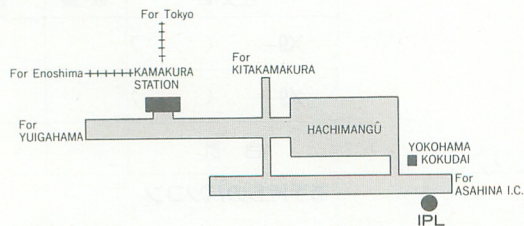
SYSTEM CREDIT

あなたのライフスタイルに合せたクレジットをどうぞ！

●ステップアップクレジットがおトク。
まず月々1,000円からスタートして2年後から3,000円へアップ。ボーナスも1年後1万円。3年後3万円。また夏のボーナスを貯金して冬のボーナスのみ1年間0円、2年後1万円、3年後2万円、また夏のボーナスのみ、又は冬のボーナスのみ年一回のお支払いもOK。さらにお支払い回数も1回払いから最長84回までご自由に設定が可能です。

●追加購入もクレジットだから便利。
追加購入も買い換えもご利用中のIPLクレジットを月々僅か1,000円ずつの調整でOK。

●ボーナスの支払いも自由自在
今から冬のボーナス払いもOK！
夏冬のボーナス2回払いも可能です。ボーナス月（6・7・8・12・1）も自由に指定、さらに、2～3年後から開始のボーナス払いもOK！

●ゆとりの8ヶ月先のお支払い！
もちろん商品はすぐお手元に。お支払いは翌月一括から8ヶ月先まで自由に設定。

IPL 株式会社アイピーエル
本社：〒248 鎌倉市雪の下4-1-12
TEL0467-24-7511 FAX0467-24-0561

株式会社 アイピーエル

本社：〒248 鎌倉市雪の下4-1-12 商品管理部：鎌倉市雪の下3-4-23
TEL. 0467-24-7511 FAX. 0467-24-0561 TEL. 0467-24-1154

●本 社 ⇒ **0467-24-7511** 電話受付：AM10：00～PM8：00（水曜定休日）FAX受付：24時間受付

●大 阪 ⇒ **06-311-2736**
●銀 座 **03-541-3058** ●仙 台 **022-266-0531**
●青 山 **03-470-0061** ●広 島 **082-293-7881**
●札 幌 **011-621-1444** ●福 岡 **092-481-2544**

●IPL商品管理部（納期、配達日のお問合せ、ご指定日のご連絡）……**0467-24-1154**
●IPLメンテナンス部（ハード上のご相談、お問合せ、初期不良の対応）……**0467-24-0453**
●IPLFAX（ご注文、お見積り、カタログ編集などスピーディに）……**0467-24-0561**
●IPLご注文お問合せ……**0467-24-1154**
●IPLビギナーズホットライン（初心者の方々への無料相談窓口）·**0467-24-0941**
●IPL テクニカルホットライン（システムアップ、買替えなどの下取り相談窓口）……**0467-24-2040**

システムを知的にコーディネイト

購入してから安心してお使いになれるように、IPLでは通信教育講座を無料で開設。添削テスト問題付で、より深く、きめ細かなアフターフォローが特徴です。

SHARP

X68000 EXPERT

アクセス No.X0901
価格 ¥864,000➡超特価 CALL!!

- CZ-602C（メインメモリ2MBマウス・トラックボール付FM音源8重和音）……¥356,000
- CZ-602D（39ミリ、アナログ3モード、チルトスタンド付き）……¥ 99,800
- EW & E1（漢字変換フロントプロセッサ搭載、高速日本語ワープロ）……¥ 38,000
- CZ-213MS（MUSIC PRO 68K）……¥ 18,8C0
- CZ-214MS（SOUND PRO 68K）……¥ 15,800
- CZ-215MS（AD PCM機能をサポートしたサンプリングエディタ）……¥ 17,800
- CZ-6BM1（MIDIボード）……¥ 26,800
- MT-32……¥ 64,000
- CZ-247MS（MIDI楽器演奏が楽しめるMUSIC PRO 68K MIDI）……¥ 28,800
- CZ-221HS（NEW Print Shop様々なカードなどを自由に作成）……¥ 19,800
- CZ-6VT1（カラーイメージユニット、テロッパー機能付き）……¥ 69,800
- Z's STAFF PRO 68K（グラフィックツール）……¥ 58,000
- CZ-8NJ2（ますますゲームがおもしろくなるサイバースティック、アナログ）……¥ 23,800
- アフターバーナー68K（サイバースティック対応）……¥ 9,200
- ロードス島戦記……¥ 9,8000
- CZ-232AS（熱血高校ドッジボール部）……¥ 7,800
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥5,000	×72回	ボーナス	47,000×12回
¥ 9,500	×72回	ボーナス	20,000×12回
¥10,000	×60回	ボーナス	29,200×10回
¥14,600	×48回	ボーナス	20,000× 8回
¥18,100	×36回	ボーナス	30,000× 6回

アクセス No.X0902
価格 ¥858,600➡超特価 CALL!!

- CZ-602C（メインメモリ2MBマウス・トラックボール付FM音源8重和音）……¥356,000
- CU-21CD（迫力の21″カラーアナログCRT3モードマルチスキャン方式）……¥139,800
- CZ-6VT1（カラーイメージユニット、テロッパー機能付き）……¥ 69,800
- Z's STAFF PRO 68K（グラフィックツール）……¥ 58,000
- CZ-223CS（フルスクリーンエディッタ内蔵の通信ソフト）……¥ 19,800
- PV-A2400MNP4（2400/1200/300bps全二重モデム クラス4）……¥ 46,800
- CZ-214MS（SOUND PRO 68K）……¥ 15,800
- CZ-221HS（NEW Print Shop様々なカードなどを自由に作成）……¥ 19,800
- CZ-8PC4（美しい印字／48ドットカラー年賀状、暑中見舞、各種カード等）……¥ 99,800
- CZ-8NJ2（ますますゲームがおもしろくなるサイバースティック、アナログ）……¥ 23,800
- アフターバーナー68K（サイバースティック対応）……¥ 9,200
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥3,900	×72回	ボーナス	50,000×12回
¥ 8,100	×72回	ボーナス	25,000×12回
¥10,000	×60回	ボーナス	25,000×10回
¥13,700	×48回	ボーナス	20,000× 8回
¥18,600	×36回	ボーナス	20,000× 6回

X68000 EXPERT HD

アクセス No.X0903
価格 ¥753,400➡超特価 CALL!!

- CZ-612C（メインメモリ2MB、40MBHDD、FM音源トラックボール付）……¥466,000
- CZ-602D（0.39ドット15型カラーディスプレイTV）……¥ 99,800
- 3M ブランクディスケット（5″ 2HD＊10枚）……¥ 18,000
- CZ-8PC4（美しい印字／48ドットカラー年賀状、暑中見舞、各種カード等）……¥ 99,800
- CZ-6VT1（カラーイメージユニット、テロッパー機能付き）……¥ 69,800
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥5,000	×72回	ボーナス	33,500×12回
¥ 9,000	×60回	ボーナス	20,000×10回
¥ 9,800	×48回	ボーナス	30,000× 8回
¥15,000	×36回	ボーナス	23,800× 6回
¥10,600	×72回	ボーナス	な し

アクセス No.D0903
価格 ¥137,800➡超特価 CALL!!

- CZ-6VT1（カラーイメージユニット、テロッパー機能付き）……¥ 69,800
- C-TRACE（CGアニメーションソフト）……¥ 68,000

月々	回数	ボーナス	
¥2,000	×72回	ボーナス	な し

X68000 PRO

アクセス No.X0904
価格 ¥797,200➡超特価 CALL!!

- CZ-652C（メインメモリ1MB、FM音源8重和音65536色マウス同梱）……¥298,000
- CZ-602D（0.39ドット15型カラーディスプレイTV）……¥ 99,800
- PC-98DO（98モード＋88モード、サウンド機能内蔵のニューマシン！）……¥298,000
- PC-CA402（アナログCRTケーブル 15ピン<—>15ピン）……¥ 4,800
- AP-550PC（24ピンカラー漢字プリンタはがきB4縦可）……¥ 69,800
- シャープX1用ケーブル……¥ 8,800
- 3M ブランクディスケット（5″ 2HD＊10枚）……¥ 18,000
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥5,000	×72回	ボーナス	35,500×12回
¥ 8,000	×60回	ボーナス	28,000×10回
¥10,200	×48回	ボーナス	30,000× 8回
¥19,600	×36回	ボーナス	な し
¥11,000	×72回	ボーナス	な し

アクセス No.X0905
価格 ¥520,200➡超特価 CALL!!

- CZ-652C（メインメモリ1MB、FM音源8重和音65536色マウス同梱）……¥298,000
- CU-21CD（迫力の21″カラーアナログCRT3モードマルチスキャン方式）……¥139,800
- CZ-8NJ2（ますますゲームがおもしろくなるサイバースティック、アナログ）……¥ 23,800
- アフターバーナー68K（サイバースティック対応）……¥ 9,200
- 大海令……¥ 12,800
- こはく色の遺言……¥ 9,800
- Super大戦略68K……¥ 8,800
- 3M ブランクディスケット（5″ 2HD＊10枚）……¥ 18,000
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥2,200	×72回	ボーナス	30,000×12回
¥ 5,000	×60回	ボーナス	20,000×10回
¥ 5,000	×48回	ボーナス	30,000× 8回
¥ 9,600	×36回	ボーナス	20,000× 6回
¥ 7,200	×72回	ボーナス	な し

アクセス No.D0902
価格 ¥33,000➡超特価 CALL!!

- CZ-8NJ2（ますますゲームがおもしろくなるサイバースティック、アナログ）……¥ 23,800
- アフターバーナー68K（サイバースティック対応）……¥ 9,200

月々	回数	ボーナス	
¥1,400	×24回	ボーナス	な し

X68000 PRO HD

アクセス No.X0906
価格 ¥804,700➡超特価 CALL!!

- CZ-662C（メインメモリ1MB、40MBHDD、FM音源8重和音マウス付）……¥408,000
- CZ-612D（0.31ドット15型カラーディスプレイTV）……¥119,800
- CZ-6BE1A（1MB増設RAMボード/CZ-601、611用）……¥ 38,000
- CZ-8PC4（美しい印字／48ドットカラー年賀状、暑中見舞、各種カード等）……¥ 99,800
- CZ-226BS（ワープロ機能を備えたカード型リレーショナルデータベース）……¥ 29,800
- CZ-221HS（NEW Print Shop様々なカードなどを自由に作成）……¥ 19,800
- CZ-6VT1（カラーイメージユニット、テロッパー機能付き）……¥ 69,800
- CZ-8NJ1（ジョイカード）……¥ 1,700
- 3M ブランクディスケット（5″ 2HD＊10枚）……¥ 18,000
- 見てみよう触れてみよう（C-TRACE 3次元CGアニメーションDEMOソフト）……プレゼント中
- X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）……★
- 初期不良期間（ワイドに1ヶ月間の交換システム！）……★
- 安心の3倍保証（IPL保証書付き）……★

月々	回数	ボーナス	
¥5,000	×72回	ボーナス	39,330×12回
¥ 8,000	×72回	ボーナス	21,300×12回
¥10,100	×60回	ボーナス	20,000×10回
¥10,000	×48回	ボーナス	36,500× 8回
¥15,800	×36回	ボーナス	30,000× 6回

限定10台！

X68000 ACE HD

CZ-611Cを10台限り特別価格にて提供いたします。20MBのハードディスクを搭載してPROの定価よりも安く！在庫をお確めの際は色指定(黒・グレー)をお忘れなく。

X68000シリーズ

EXPERT	定価~~¥356,000~~
EXPERT HD	定価~~¥466,000~~
PRO	定価~~¥298,000~~
PRO HD	定価~~¥408,000~~

各シリーズとも特価販売中！
T・ZONE 2Fにて。

ADO・TOYOMURA T・ZONE ティー・ゾーン

Micom Zone

2F 〒101 東京都千代田区外神田4-4-1 ☎257-2650

グレードアップセール
8月1日〜8月31日

☆この広告の提示価格には、消費税は含まれておりません。

X68000をトータルサポート
T・ZONE 2F

SHARP Authorized………

X68000 PRO SHOP

お久し振りです。故あってしばらくお休みしましたが(広告を、です)。皆様おかわりございませんでしょうか。そうですか。それはようございました。この短い間にも、思えば様々なできごとがございました。TOWNSの発表にそぞろ心乱れる日のなかったでもない。J3100SSに熱き思いを寄せたこともしばし。この短かき充電期を糧として皆様と共にX68000の未来を考え、また共に気を揉み続けてゆこうと思う今日この頃です。では、

店主拝

OS-9/68000 for X68000

□OS9/68000 (SHARP) ¥29,800
□C＆PRO PACK(マイクロウェア) ¥58,000
□Src Dbg (マイクロウェア) ¥39,800
□MW-BASIC(マイクロウェア) ¥60,000
□BTree09 (ARK) ¥36,000
MW-BASIC用のISAM用B-Treeパッケージです。応用例として住所録と販売管理プログラムが付属。全ソースコード付です。(このソフトを動かすためにはMW-BASICが必要です。)
□UD-CACHE(ARK) ¥16,000
すべてのRBFデバイスに対応するキャッシュです。
□FBU(ARK) ¥38,000
ハード・ディスクバックアップユーティリティーです。巨大ファイルを分割バックアップしたり、日付管理を行なったバックアップもOK。
□VSED(FORKS) ¥28,000
OS9/68000で唯一オートバッファリングをサポートしたスクリーンエディタです。
□CSG IMS……は今対応中です。もう少々お待ち下さい。

SHARP Authorized………

増設OK CZ-620H

SHARP純正20MBHD

効能 ①HD内蔵タイプのX68000に増設可。
②すでにHDを接続していても増設可。
(シャープ以外のハードディスクの場合でもご相談下さい。)
③もちろん最初の1台としても安心。
④なかなかサマになるデザインです。

限定！
定価¥178,000⇒*Special Price!*

IO-730

定価¥230,000

X68000のカラー機能をフルに活かす。フルカラーインクジェットプリンタの本命、特別価格にて

T・ZONE正社員・長期アルバイト募集中！
☆お問い合わせは総務課鈴木まで(TEL 03-257-2630)

8F オフィス
7F 工学書輸入書
6F 衛星放送
5F HAM トランシーバー
4F トレードゾーン
3F IBMゾーン
2F パソコンソフトウェア
1F パーソナルOA
B1 Macintosh!

T・ZONE
営業時間：AM10:30〜PM7:00

下記T・ZONE各店でも扱っています。

宇都宮店：☎0286(63)4949 川口店：☎0482(68)7826 ラジオショップ：☎03(257)2643 横浜店：☎045(641)7741
大宮店：☎048(652)1831 東ラジ店：☎03(257)2694 パーツショップ：☎03(257)2655 静岡店：☎0542(83)1331

●マイコン通販利用の方へ：現金書留で送金される際は、住所、氏名、TEL番号、希望商品名(詳しく)を明記して下さい 振込を御希望の方は下記銀行へお願いします
尚、いずれも予めTELにて、御予約・送料確認の上御送金下さい (振込口座 埼玉銀行 秋葉原支店 当座2705 株亜土電子工業)

ACCESS

X1エミュレータ 定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3～5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

ファイル転送ユーティリティ

ディスク転送

X1ディスク⟷X68000 Human68k (5"2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC:CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

実行可能アプリケーションソフト

●HuBASIC ●X1 CP/M ●X1 LOGO

[X1 CP/M用ランゲージシリーズ] ●APL ●LISP ●COBOL ●C ●FORTH ●FORTRAN ●PASCAL

●etc (X1シリーズ用とされているものに限ります。)

*プロテクトの施してあるソフトは実行できません。
*一部サポートしていない機能があります。
*タイミング等ハードウェアに依存するようなものは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
*turbo専用のソフトは動作致しません。

MS-DOSエミュレータ CONCERTO-X68K 定価¥99,800

CONCERTO(コンチェルト)-X68KはX68000上でお使い頂くMS-DOSエミュレータ(専用ハード+ソフトウェア)です。特定機種用と限定されていないMS-DOS(V2.11)用のソフトがX68000上でお使い頂けます。MS-DOSソフトの実行は、NEC V30CPUを使用した専用ハードウェア(DOS Engine)を利用するため高速実行を実現しております。

ベンチマークテストの結果を見て頂いてもわかるように、PC-9801上で実行するよりもX68000上で実行する方が高速に処理できることを確認しております。

MS-C(4.00)を用いてベンチマークテスト

マシン:X68000 ACEHD　:PC-9801VM(V30)
比較条件:CONCERTO-X68K　:MS-DOS V2.11
　フロッピーディスクを使用:フロッピーディスクを使用
実験方法:FILES=20
　CONCERTO-X68K側ではMS-DOS V2.11に含まれるCOMMAND.COM上よりコンパイラを起動
　^Cを入力しバッファをクリアした後バッチジョブを実行

*実験マシンは共にRAMDISK、8087等は使用しておりません。

◆SAVAGE.C
(三角関数、対数関数、平方根関数の演算速度と精度をテストするためのプログラム)

	CONCERTO-X68K	PC-9801VM(10MHz)	PC-9801VM(8MHz)
コンパイル時間+LINK	93	175	174
実行時間	77	78	96

◆SIEVE.C (エラトステネスのふるいプログラム)

	CONCERTO-X68K	PC-9801VM(10MHz)	PC-9801VM(8MHz)
コンパイル時間+LINK	67	119	121
実行時間	116	119	148

(単位は秒　時間計測用プログラムを含む)

```
A>XDOSINIT↵ ------- エミュレータ起動時に必要な初期設定
                     通常はじめに1回だけの実行で可
ＣＯＮＣＥＲＴＯ－Ｘ６８Ｋ Ver 1.00 Copyright (C) 1988 ACCESS CO.,LTD.

 アドレス 00BE0000 に使用できるＤＯＳ Ｅｎｇｉｎｅがあります。

 ＣＯＮＣＥＲＴＯ－Ｘ６８Ｋを初期設定中です。

  使用可能なメモリサイズは 512 キロバイトです。        共有メモリ,ハンドシェイク
  ＤＯＳ Ｅｎｇｉｎｅからの割り込みレベルは 2 です。  割り込み等のチェック
  ８０８７は実装されていません。

 ＣＯＮＣＥＲＴＯ－Ｘ６８Ｋが使用可能です。

A>XDOS 〈コマンド〉〈パラメータ〉↵ ------- コマンドはMS-DOSソフト名、パラメータはそのソフトが
                                          必要とするパラメータの並び
 または                                   実行終了後、制御はHuman68kに戻る
A>XDOS COMMAND↵ --------- COMMAND.COM起動後はMS-DOSの環境として使用可

Command バージョン 2.11

XDOS:A>〈コマンド〉〈パラメータ〉↵ --------- 実行終了後も制御はそのまま

XDOS:A>EXIT↵ ----------- CONCERTO-X68Kを抜けてHuman68kに戻る
```

(CONCERTO-X68Kの実行、下線部はキー入力)

専用ハード:DOS Engine

- 8MHzのV30を使用(メモリノーウェイト)
- ボード上にMS-DOSの実行用メモリ512KByte搭載
- 数値演算プロセッサ8087-1実装可能(オプション)

*ボードは本体より12cm程度大きくなります。その部分にはカバーがつきます。

MS-DOS用実行可能アプリケーションソフト

- MS-C(Ver3.00, 4.00)
- MS-FORTRAN(Ver3.13, 4.01)
- MS-PASCAL(Ver3.13)
- MS-LINK(Ver2.01, 2.20, 2.44)
- MS-BASIC(Ver5.27)
- Lattice C(Ver2.12, 3.10)
- Optimizing-C(Ver2.20F)
- TURBO PASCAL(Ver2.00B, 3.01A)
- Plink 86(Ver1.46)
- etc……

(実行可能ソフトの一例です。)

代理店募集

アクセスではこれらの製品の発売にあたり代理店を募集しております。詳しくはお問い合せください。

*この商品の価格には消費税は含まれておりません。
*MS-DOSはマイクロソフト社, CP/Mはデジタルリサーチ社の商標です。
COMMAND.COMはMS-DOSに標準のコマンドプロセッサです。上記のソフトウェアは各社の商標です。
*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス　〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(233)0200(代) FAX.03(291)7019

資料請求券 oh!X 9月号

ADVANCED TURBO

先駆の"Z"アビリティがパソコンクリエイターを魅了する。

X1 パソコンテレビ turbo Z III

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK	標準価格 169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格 92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格 5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロッパ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV, RGB, HALF, CDOWN, CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準準拠漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。

資料請求券 X1turbo ZIII Oh!X 9月号

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545·大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税をお支払い下さい。

T4910217909569 雑誌02179-9

Oh!X 9月号 1989年9月1日発行(毎月1回1日発行) 通巻第89号 昭和58年11月2日第三種郵便物認可 ㈱日本ソフトバンク発行 Printed in Japan 定価560円(本体544円)